

(11)Publication number : **2000-299641**
(43)Date of publication of application : **24.10.2000**

(21)Application number : **11-104162**
(22)Date of filing : **12.04.1999**
(71)Applicant : **MITSUBISHI ELECTRIC CORP**
(72)Inventor : **KIMURA TOMOHIRO
YOSHIDA MASAYUKI
ONO FUMITAKA**

図1 システムバス

CPU ROM

RAM ディスクドライブ

テープドライブ

プリンター キーボード

電源 接地

SA

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-299641

(P 2 0 0 0 - 2 9 9 6 4 1 A)

(43) 公開日 平成12年10月24日 (2000. 10. 24)

(51) Int. Cl. ⁷

識別記号

F I

テーマコード (参考)

H03M 7/40
G06F 5/00
H04N 1/413
7/24

H03M 7/40
G06F 5/00
H04N 1/413
7/13

5C059
H 5C078
Z 5I064
Z 9A001

審査請求 未請求 請求項の数29 O L (全36頁)

(21) 出願番号

特願平11-104162

(22) 出願日

平成11年4月12日 (1999. 4. 12)

(71) 出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 発明者 木村 智広

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

(72) 発明者 吉田 雅之

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

(74) 代理人 100057874

弁理士 曾我 道照 (外6名)

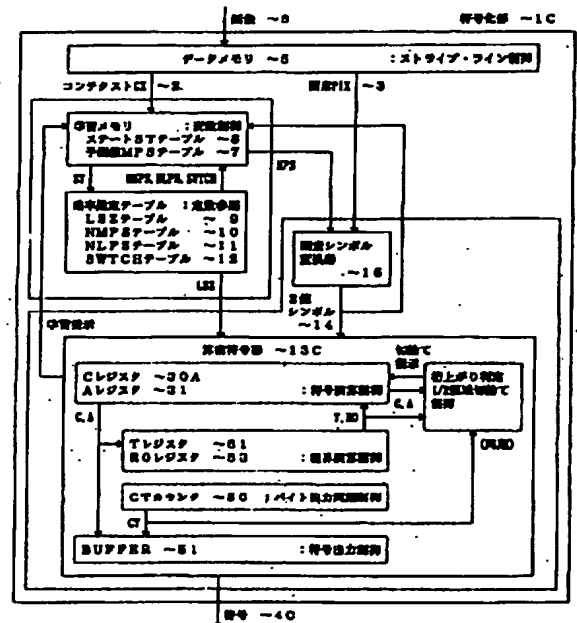
最終頁に続く

(54) 【発明の名称】 符号化装置、復号化装置、符号化復号化装置、符号化方法並びに復号化方法

(57) 【要約】

【課題】 桁上りを制御により符号を確定させ所定のパターンへの置き換えにより総符号長と符号の掃出し時間を短縮する。

【課題手段】 情報源データを蓄積するデータメモリ、補助パラメータを基に指定される符号化対象データに関する学習データを蓄積する学習メモリ、学習データを基に指定される符号化パラメータを出力する確率推定テーブル、符号化対象データと符号化パラメータを基に算術符号化を行い符号を出力する符号器を備える符号化装置において、情報源データの入力処理毎または符号の出力毎に所定の間隔を計測して通知する同期検出器、所定の間隔で有効領域内の桁上り境界値を検出すると共に、その検出結果に基づいて有効領域内の切り捨て領域を指示する桁上り境界検出器を備え、符号器は、桁上り検出器により指示された有効領域を上位と下位に等分した部分領域の一方を切り捨てて有効領域を更新する。



【特許請求の範囲】

【請求項1】 情報源データを蓄積して符号化対象データとその補助的なパラメータ（コンテキスト）を出力するデータメモリと、上記補助パラメータを基に指定される上記符号化対象データに関する学習データを蓄積し出力する学習メモリと、上記学習データを基に指定される符号化パラメータを出力する確率推定テーブルと、上記符号化対象データと上記符号化パラメータを基に算術符号化を行い符号を出力する符号器とを備える符号化装置において、

所定の単位の情報源データの入力処理または符号の出力を所定の間隔として計測して通知する同期検出器と、上記所定の間隔で有効領域内の桁上がり境界値を検出すると共に、その検出結果に基づいて有効領域内の切り捨て領域を指示する桁上がり境界検出器とを備え、上記符号器は、上記桁上がり検出器により指示された上記有効領域を上位と下位に等分した部分領域の一方を切り捨てて上記有効領域を更新することを特徴とする符号化装置。

【請求項2】 上記符号器は、上記桁上がり境界検出器で桁上がり境界が検出されたとき、上記桁上がり境界検出器により指示された上記有効領域を上位と下位に等分した部分領域の一方を切り捨てて上記有効領域を更新することを特徴とする請求項1に記載の符号化装置。

【請求項3】 上記桁上がり境界検出器は、検出された桁上がり境界を含む部分領域を切り捨てるように指示することを特徴とする請求項1または2に記載の符号化装置。

【請求項4】 上記所定の間隔は、所定単位の符号化対象データを符号化したときであることを特徴とする請求項1または2に記載の符号化装置。

【請求項5】 上記所定の間隔は、符号値の確定まで待機させる符号長の最大値を基準に設定されることを特徴とする請求項4に記載の符号化装置。

【請求項6】 上記符号器は、上記所定の間隔を基準に符号値の確定まで待機させる符号長の最大値が設定されることを特徴とする請求項4に記載の符号化装置。

【請求項7】 上記所定の間隔は、所定単位の符号出力が発生したときであることを特徴とする請求項1または2に記載の符号化装置。

【請求項8】 上記所定の間隔は、所定単位の符号出力が発生し、かつ符号出力値が所定値となるときであることを特徴とする請求項1または2に記載の符号化装置。

【請求項9】 上記桁上がり境界検出器は、上記有効領域内に桁上がり境界が検出されなくても上記有効領域を上位と下位に等分した部分領域のどちらか一方の部分領域を指示することを特徴とする請求項1に記載の符号化装置。

【請求項10】 上記符号器は、確定した符号に同じ値が連続するときにその値と長さを特定できるランレン

グスマーカに置き換えるランレングスマーカ変換器を有することを特徴とする請求項1または2に記載の符号化装置。

【請求項11】 上記ランレングスマーカ変換器は、置き換えずにそのまま出力した方が変換される上記ランレングスマーカより符号出力が短いときには変換を行わないことを特徴とする請求項10に記載の符号化装置。

【請求項12】 上記ランレングスマーカから抽出される連続長の最大値は、復号の際のランレングスマーカ逆変換の結果に基づいて上記符号器が上記ランレングスマーカ変換器に設定することを特徴とする請求項10に記載の符号化装置。

【請求項13】 復号対象データに対する補助的なパラメータ（コンテキスト）を出力し復号された上記復号対象データを蓄積して情報源データとして出力するデータメモリと、上記補助パラメータを基に指定される上記復号対象データに関する学習データを蓄積し出力する学習メモリと、上記学習データを基に指定される復号パラメータを出力する確率推定テーブルと、上記復号パラメータと符号を基に算術復号を行い上記復号対象データを出力する復号器とを備える復号化装置において、

所定の単位の情報源データの出力処理または符号の入力を所定の間隔として計測して通知する同期検出器と、上記所定の間隔で有効領域内の桁上がり境界値を検出すると共に、その検出結果に基づいて有効領域内の切り捨て領域を指示する桁上がり境界検出器とを備え、上記復号器は、上記有効領域を上位と下位に等分した部分領域の上記符号値を含まない一方を切り捨てて上記有効領域を更新することを特徴とする復号化装置。

【請求項14】 上記復号器は、上記桁上がり境界検出器で桁上がり境界が検出されたとき、上記有効領域を上位と下位に等分した部分領域の上記符号値を含まない一方を切り捨てて上記有効領域を更新することを特徴とする請求項13に記載の復号化装置。

【請求項15】 上記桁上がり境界検出器は、検出された桁上がり境界を含む部分領域を切り捨てるように指示することを特徴とする請求項13または14に記載の復号化装置。

【請求項16】 上記所定の間隔は、所定単位の復号対象データを復号したときであることを特徴とする請求項13または14に記載される復号化装置。

【請求項17】 上記所定の間隔は、所定単位の符号入力が発生したときであることを特徴とする請求項13または14に記載される復号化装置。

【請求項18】 上記所定の間隔は、所定単位の符号入力が発生し、かつ復号と同時に再現される符号出力値が所定値となるときであることを特徴とする請求項13に記載の復号化装置。

【請求項19】 上記復号器は、ランレングスマーカを検出すると、置き換えられた長さの元の符号値に逆変換

するランレングスマーカ逆変換器を有することを特徴とする請求項 1 3 または 1 4 に記載される復号化装置。

【請求項 2 0】 符号化装置として、情報源データを蓄積して符号化対象データとその補助的なパラメータ（コンテキスト）を出力するデータメモリと、上記補助パラメータを基に指定される上記符号化対象データに関する学習データを蓄積し出力する学習メモリと、上記学習データを基に指定される符号化パラメータを出力する確率推定テーブルと、上記符号化対象データと上記符号化パラメータを基に算術符号化を行い符号を出力する符号器と、所定の単位の情報源データの入力処理または符号の出力を所定の間隔として計測して通知する同期検出器と、上記所定の間隔で有効領域内の桁上がり境界値を検出すると共に、その検出結果に基づいて有効領域内の切り捨て領域を指示する桁上がり境界検出器とを備え、上記符号器は、上記桁上がり検出器により指示された上記有効領域を上位と下位に等分した部分領域の一方を切り捨てて上記有効領域を更新すると共に、

復号化装置として、復号対象データに対する補助的なパラメータ（コンテキスト）を出力し復号された上記復号対象データを蓄積して情報源データとして出力するデータメモリと、上記補助パラメータを基に指定される上記復号対象データに関する学習データを蓄積し出力する学習メモリと、上記学習データを基に指定される復号パラメータを出力する確率推定テーブルと、上記復号パラメータと符号を基に算術復号を行い上記復号対象データを出力する復号器と、所定の単位の情報源データの出力処理または符号の入力を所定の間隔として計測して通知する同期検出器と、上記所定の間隔で有効領域内の桁上がり境界値を検出すると共に、その検出結果に基づいて有効領域内の切り捨て領域を指示する桁上がり境界検出器とを備え、

上記復号器は、上記有効領域を上位と下位に等分した部分領域の上記符号値を含まない一方を切り捨てて上記有効領域を更新することを特徴とする符号化復号化装置。

【請求項 2 1】 上記符号器は、上記桁上がり境界検出器で桁上がり境界が検出されたとき、上記桁上がり境界検出器により指示された上記有効領域を上位と下位に等分した部分領域の一方を切り捨てて上記有効領域を更新すると共に、上記復号器は、上記桁上がり境界検出器で桁上がり境界が検出されたとき、上記有効領域を上位と下位に等分した部分領域の上記符号値を含まない一方を切り捨てて上記有効領域を更新することを特徴とする請求項 2 0 に記載の符号化復号化装置。

【請求項 2 2】 (a) 情報源データを蓄積して符号化対象データとその補助的なパラメータ（コンテキスト）を出力するデータ蓄積ステップと、(b) 上記補助パラメータを基に指定される上記符号化対象データに関する学習データを蓄積し出力する学習ステップと、(c) 上記学習データを基に指定される符号化パラメータを出力す

る確率推定ステップと、(d) 上記符号化対象データと上記符号化パラメータを基に算術符号化を行い符号を出力する符号化ステップと、(e) 所定の単位の情報源データの入力処理または符号の出力を所定の間隔として計測して通知する同期検出ステップと、(f) 上記所定の間隔で有効領域内の桁上がり境界値を検出すると共に、その検出結果に基づいて有効領域内の切り捨て領域を指示する桁上がり境界検出ステップと、(g) 上記桁上がり境界検出ステップにより指示された上記有効領域を上位と下位に等分した部分領域の一方を切り捨てて上記有効領域を更新する符号化修正ステップとを有する符号化方法。

【請求項 2 3】 上記符号化修正ステップは、上記桁上がり境界検出ステップで桁上がり境界が検出されたとき、上記桁上がり境界検出ステップで指示された上記有効領域を上位と下位に等分した部分領域の一方を切り捨てて上記有効領域を更新することを特徴とする請求項 2 2 に記載の符号化方法。

【請求項 2 4】 上記符号化ステップおよび上記符号化修正ステップは、確定した符号に同じ値が連続するとき、その値と長さを特定できるランレングスマーカに置き換える変換ステップを有することを特徴とする請求項 2 2 または 2 3 に記載の符号化方法。

【請求項 2 5】 上記変換ステップは、置き換えずにそのまま出力した方が変換される上記ランレングスマーカより符号出力が短いときには変換を行わないことを特徴とする請求項 2 4 に記載の符号化方法。

【請求項 2 6】 上記ランレングスマーカから抽出される連続長の最大値は、復号の際、ランレングスマーカ逆変換の結果に基づいて上記符号化ステップが上記変換ステップに設定することを特徴とする請求項 2 4 に記載の符号化方法。

【請求項 2 7】 (a) 復号対象データに対する補助的なパラメータ（コンテキスト）を出力し復号された上記復号対象データを蓄積して情報源データとして出力するデータ蓄積ステップと、(b) 上記補助パラメータを基に指定される上記復号対象データに関する学習データを蓄積し出力する学習ステップと、(c) 上記学習データを基に指定される復号パラメータを出力する確率推定ステップと、(d) 上記復号パラメータと符号を基に算術復号を行い上記復号対象データを出力する復号化ステップと、(e) 所定の単位の情報源データの出力処理または符号の入力を所定の間隔として計測して通知する同期検出ステップと、(f) 上記所定の間隔で有効領域内の桁上がり境界値を検出すると共に、その検出結果に基づいて有効領域内の切り捨て領域を指示する桁上がり境界検出ステップと、(g) 上記桁上がり境界検出ステップにより指示された上記有効領域を上位と下位に等分した部分領域の一方を切り捨てて上記有効領域を更新する復号化修正ステップとを有する復号化方法。

【請求項 28】 上記復号化修正ステップは、上記桁上がり境界検出ステップで桁上がり境界が検出されたとき、上記桁上がり境界検出ステップで指示された上記有効領域を上位と下位に等分した部分領域の一方を切り捨てて上記有効領域を更新することを特徴とする請求項 27 に記載の復号化方法。

【請求項 29】 上記復号化ステップは、ランレングスマーカを検出すると、置き換えられた長さの元の符号値に逆変換するランレングスマーカ逆変換ステップを有することを特徴とする請求項 27 または 28 に記載の復号化方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 この発明は、画像あるいはデータの算術符号化および算術復号化に係るもので、符号化装置、復号化装置、符号化復号化装置、符号化方法並びに復号化方法に関するものである。

【0002】

【従来の技術】 図 39 は算術符号化の概念を示す説明図である。図 39 において、数直線上の全体の有効領域を優勢シンボル MPS と劣勢シンボル LPS に対する二つの部分領域に分割し、出現したシンボルに対応する部分領域を新規有効領域に更新するという手順で符号化が行われる。しかし、領域幅が小さくなるにつれて 2 進小数（固定小数点）で示したように下界値、上界値は有効桁数が増加し、演算のためのレジスタ精度も必要となってくる。

【0003】 ここで、有効桁数が増加しても上位桁はほとんど変化がないことに注目し、図 40 に示すように、小数部を常に領域分割の精度（図では 3 桁）に保ち、上位桁を整数部に掃き出すために正規化処理を行う。図 40 では、正規化処理で 2 のべき乗倍に拡大することによって、有効領域幅 A を $1/2$ に当たる 4 以上、1 に当たる 8 以下（8 は初期値のみ）にとし、下界値（符号）C とすれば有効領域は C 以上 C + A 未満で表すことができる。

【0004】 第 1 の従来例について。

第 1 の従来例の算術符号化の符号器（Encoder）・復号器（Decoder）は、ITU-T の国際標準勧告 T. 82 に記載されるテーブルおよび処理を説明するフローチャートによって実現できる。以下、この算術符号化を QM-Coder、その符号器を符号部 1A、その復号器を復号部 16A と呼ぶものとし、その符号器と復号器の概略構成を図 41、図 42 に示す。

【0005】 図 41 において、符号部 1A の入力 2 つあり、第 1 の入力はコンテキスト CX（Context）2 であり、第 2 の入力は符号化する画素 PIX（Pixel）3 である。出力は符号（Code）4A である。また、図 42 において、復号部 16A の入力 2 つあり、第 1 の入力はコンテキスト CX2 であり、第 2 の入力は符号 4A で

ある。出力は、復号された画素 PIX3 である。

【0006】 データメモリ 5 は、画像（Image）6 を蓄積し、符号化または復号化の対象となる画素に対して共通に参照できる近隣のすでに符号化／復号化済みの画素の中で、参照位置をモデルテンプレート（Model Template）で指定される 10 画素の参照パターンであるコンテキスト（10 ビット。総数 1024）CX2 を生成する。また、符号化のときは同時に符号化対象画素を出力し、復号化のときは復号された復号対象画素を蓄積する。

【0007】 QM-Coder には、コンテキスト CX2 を生成するための図 43 に示される 2 ラインおよび 3 ラインの標準モデルテンプレートのいずれかを選択し、規定のヘッダ情報によって符号器から復号器に通知する。

【0008】 QM-Coder では、符号化／復号化対象画素に対する各コンテキストごとに画素値の予測一致確率を推定し、その変動に伴って学習しながら符号化／復号化を進める。学習は、コンテキストをインデックスとする 2 つの変数テーブルで構成される学習メモリを書き換えることによって行われる。1 つは、出現する確率の高い画素値 MPS（More Probable Symbol：優勢シンボル）を予測値として記憶する各 1 ビットの MPS テーブル 7 である（予測一致確率の低い画素値は LPS [Less Probable Symbol：劣勢シンボル] という）。もう 1 つは、予測値の予測一致確率の程度を総計 113 の状態（State）に分類した状態番号（0～112）を記憶する各 7 ビットの ST テーブル 8 である。初期値は MPS テーブル 7、ST テーブル 8 とともに全て 0 である。

【0009】 変数テーブルの他に、符号化／復号の際に状態番号（ステート）をインデックスとして参照する 4 つの定数テーブルで構成される確率推定テーブル（Probability Estimation Table）が存在する。LPS 領域幅を 16 ビットで表す LSZ テーブル 9、MPS 遷移先を 7 ビットで表す NMPS テーブル 10、LPS 遷移先を 7 ビットで表す NLPS テーブル 11、予測値反転判定を 1 ビットで表す SWITCH テーブル 12 である。その設定値を図 44 に示す（ここで示した英字の変数・定数テーブル名は、後述の処理を説明するフローチャートで用いる配列名とする）。

【0010】 LSZ テーブル 9 は、算術符号器 13A／算術復号器 17A 内の演算部で参照され、適応予測の学習には直接的には関係しない。算術符号器 13A／算術復号器 17A の内部では LSZ 値を用いて演算が行われ、演算精度が低下した際には正規化処理（Renormalization）を行う。この正規化処理が行われるとき、同時に学習が指示される。

【0011】 符号化のとき、画素シンボル変換器 15 は、コンテキスト CX2 をインデックスとして MPS テーブル 7 から出力された MPS 値と画素 PIX3 の一致／

不一致を判定して、2値シンボル14を算術符号器13A、学習メモリ(MPSテーブル7、STテーブル8)へ出力し、学習指示により学習を行う。

【0012】また、復号のとき、シンボル画素変換器18は、算術復号器17Aで復号された2値シンボル14から、コンテキストCX2をインデックスとしてMPSテーブル7から出力されたMPS値と復号すべき画素PIX3の一致／不一致を判定し、一致ならばMPS値、不一致ならばLPS値(=1-MPS)を画素PIX3値ととして復号する。復号された2値シンボル14は、学習メモリ(MPSテーブル7、STテーブル8)へ出力し、学習指示により学習を行う。

【0013】学習が指示されたとき、符号化／復号対象の2値シンボル14が優勢シンボルであればNMPSテーブル値10を、劣勢シンボルであればNLPSTテーブル値11をSTテーブル8に書き込み、状態遷移が実現される。また、劣勢シンボルによる学習では、その予測一致確率が1/2であればMPSテーブル値を反転(演算「1-MPS」)させてMPSテーブルに書き込む。その一致確率が1/2であるかどうかは、SWTCH値をフラグとして判定される。

【0014】算術符号器13Aは、LSZ値9と2値シンボル14の入力から、符号(領域下界値)を示すCレジスタ30Aと領域幅を示すAレジスタ31で符号化演算が進められ、CTカウンタ50でバイト単位の符号出力タイミングを計り、BUFFER51とSCカウンタで桁上がりの伝搬可能性のある未確定の符号出力待機と確定した符号出力を行い、符号4Aを出力する。

【0015】また、算術復号器17Aは、LSZ値9と符号4Aの入力から、CTカウンタ50でバイト単位の符号入力タイミングを計り、BUFFER51を介して、領域下界値から符号4Aまでの変位を示すCレジスタ30Bと、領域幅を示すAレジスタ31で復号演算が進められ、2値シンボル14を出力する。

【0016】図41と図42に示される構成ブロック内およびブロック間の詳細な処理動作は、後述の処理を説明するフローを示す図46ないし図62で説明する。

【0017】符号化処理を説明するフローおよび復号処理を説明するフローを説明する前に、符号化レジスタ(Cレジスタ)30A、復号レジスタ(Cレジスタ)30Bおよび領域幅レジスタ(Aレジスタ)31のビット配置を図45に示す。

【0018】符号化レジスタ(C)30Aにおいて、bit15とbit16の間に小数点を設定し、"x"(16ビット)はLSZ9に対する演算部Cx32であり、キャリがある場合はより上位へ伝搬する。"s"(3ビット)はスペーサビット部Cs33、"b"(8ビット)はバイト出力部Cb34、"c"(1ビット)は桁上がり判定部Cc35である。符号化の過程において、Cレジスタ値は符号4として符号化したシンボルに対応させた領

域の下界値となるように更新される。

【0019】復号レジスタ(C)30Bにおいて、下位ワードCLOW36と上位ワードCHIGH38は32ビットのレジスタとして実現でき、MSBであるbit31の上位に小数点を設定し、"b"(8ビット)はバイト入力部(CLOWレジスタ36)の上位バイトCb37、"x"(16ビット)はLSZ9に対する演算部Cx(CHIGHレジスタ38)39である。復号の過程において、Cレジスタ値は復号したシンボルに対応させた領域の下界値からその領域内の座標である符号4へのオフセット値となるように更新される。

【0020】領域幅レジスタ(A)31は符号化／復号で共通とし、符号化／復号レジスタ30A、30Bの小数点に対応して、"x"レジスタ部に合わせて"a"(16ビット)が小数部として配置され、初期状態の値でのみ整数部(bit16)が"1"なる。領域幅(領域サイズともいう)は、A-LSZ(下方領域幅)またはLSZ(上方領域幅)に更新され、初期値(整数部="1")を除いて1/2の重みを示すbit15が"1"となるように正規化され、1/2以上に保つことによって、上方領域幅としていかなるLSZ9を選択しても下方領域の確保を保証する。正規化処理では、Aレジスタ31、Cレジスタ30Aまたは30Bを同時にシフト処理によって拡大する。

【0021】QM-Coderでは、状態に対して固定サイズとなる上方領域LSZ9は通常劣勢シンボルに割り当てられるが、下方領域が上方領域より小さくなるときには優勢シンボルに割り当てる「条件付きMPS/LPS交換」を行っている。劣勢シンボルを符号化／復号したとき、および条件付きMPS/LPS交換を適用して優勢シンボルを符号化／復号したときには正規化処理が必ず行われる。

【0022】このレジスタのビット配置を基に、符号化の処理を説明するフロー、復号の処理を説明するフローを説明する。処理を説明するフロー中の用語「レイヤ」は階層的符号化の場合の「(解像度)層」、「ストライプ」は画像をNライン単位(最終ストライプのみNライン以下、また、N=1でも構わない)で区切った「短冊」を意味する。ここでは、レイヤ数=1、ストライプ=ライン(N=1)として説明するが、複数レイヤの符号化／復号への拡張を妨げるものではない。

【0023】符号化／復号処理を説明するフローを説明するために、図41、図42、図45で説明した変数、テーブル、レジスタの他に、次の補助変数CT50、BUFFER51、SC52、TEMP53を使用する。CT50は、正規化処理によるCレジスタ30A、30BおよびAレジスタ31のシフト数を計数し、値が0となったとき次の符号バイト入出力を行うための補助変数である。BUFFER51は、符号化のときCレジスタ30Aから出力された符号バイト値、復号のときCレジ

スタ 30B へ入力される符号バイト値を格納する補助変数である。SC52 は、符号化のみで使用され、レジスタ 30A から出力された符号バイト値が 0xFF であるときその連続数を計数する補助変数である。

【0024】なお、本明細書において“0x”で始まる数値は 16 進数を表すものとする。TEMP53 は、符号化のみで使用され、BUFFER51 への桁上りを検出し、桁上り処理後はその下位 8 ビットを新たな BUFFER 値 51 とするための補助変数である。レジスタ 30A から TEMP53 を介して設定される BUFFER51 は桁上り処理を行わずに 0xFF となることはなく、その時点で BUFFER51 から下位、つまり BUFFER51、SC52 個の 0xFF はレジスタ 30A より上位への桁上りがあれば変更される可能性があるためレジスタ 30A から出力されていても符号 4 として確定できていない。

【0025】図 46 は、符号化処理の全体の流れを示す ENCODER 処理を説明するフローチャートである。国際標準勧告 T. 82 の本処理を説明するフローチャートにある TP (Typical Prediction) および DP (Deterministic Prediction) に関する処理は、直接的な関係がないため省略する。まず、S101 では INITENC 処理を呼び出し、符号化処理の初期化を行う。S102 では画素 PIX3 とコンテキスト CX2 の組を 1 つずつ読み出し、S103 では ENCODE 処理によって符号化を行う。S104 ではストライプ（ここではライン）が終了するまで S102 および S103 を繰り返すし、さらに S105 では画像が終了するまで S102 から S104 までストライプの符号化を繰り返す。S106 で FLUSH 処理を呼び出して符号化処理の後処理を行う。

【0026】図 47 は、符号化対象画素値 3 と予測値 7 の一致・不一致から呼び出す処理を切り替える ENCODE 処理を説明するフローチャートである。S111 では画素値 3 と予測値 7 の一致・不一致を判定し、一致ならば優勢シンボル、不一致ならば劣勢シンボルを符号化する。S113 では CODEMPS 処理を呼び出して優勢シンボルを、S112 では CODELPS 処理を呼び出して劣勢シンボルを符号化する。

【0027】図 48 は、符号化対象画素値 3 と予測値 7 が不一致のとき、すなわち劣勢シンボルを符号化する際に呼び出される CODELPS 処理を説明するフローチャートである。まず、S121 では一時的に A レジスタ 31 値は、下方領域幅に更新される。S122 の判定が Yes ならば、条件付き MPS/LPS 交換が適用され、下方領域を符号化するため A レジスタ 31 はそのままとし、C レジスタ 30A は更新しない。S122 の判定が No ならば、上方領域を符号化し、S123 で領域下界値である C レジスタ 30A、S124 で領域幅である A レジスタ 31 を更新する。S125 の判定で定数 S

WTCH 値 12 が 1 ならば、S126 で予測値 (MPS テーブル) の反転・更新を行う。LPS 符号化では、S127 で NLPST テーブル 11 参照による状態遷移を行い、S128 で RENORME 処理を呼び出して正規化処理を行う。

【0028】図 49 は、符号化対象画素値 3 と予測値 7 が一致のとき、すなわち優勢シンボルを符号化する際に呼び出される CODEMPS 処理を説明するフローチャートである。まず、S131 では一時的に A レジスタ 31 値は、下方領域幅に更新される。S132 の判定が No ならばそのまま CODEMPS 処理を終了する。S132 の判定が Yes ならば、必ず S136 で NMPS テーブル 10 参照による状態遷移を行い、S137 で RENORME 処理を呼び出して正規化処理を行う。S136 および S137 の前に、S133 の判定が Yes ならば、下方領域を符号化するため A レジスタ 31 はそのままとし、C レジスタ 30A は更新しない。S133 の判定が No ならば、条件付き MPS/LPS 交換が適用され、上方領域を符号化し、S134 で C レジスタ 30A、S135 で A レジスタ 31 を更新する。

【0029】図 50 は、正規化処理を行う RENORME 処理を説明するフローチャートである。S141 では A レジスタ 31、S142 では C レジスタ 30A を 1 ビット上位へシフトすることによって 2 倍の乗算と等価な演算を行う。S143 で変数 CT50 から 1 を減じ、S144 で変数 CT50 値が 0 か否かを判定し、判定が Yes ならば S145 で BYTEOUT 処理を呼び出して C レジスタ 30A から 1 バイト符号 4 を出力する。S146 の判定は、正規化処理の終了判定を行っており、A レジスタ 31 が 0x8000 未満ならば S141 から S145 を繰り返し、0x8000 以上となれば終了する。

【0030】図 51 は、C レジスタ 30A から 1 バイトずつ符号 4 を出力させる BYTEOUT 処理を説明するフローチャートである。出力するバイトは符号化レジスタ C30A のバイト出力部 Cb34 である。桁上りを判定するための桁上り判定部 Cc35 も同時に処理する。S151 で Cb レジスタ 34 および Cc レジスタ 35 の計 9 ビットを変数 TEMP53 に設定する。S152 で桁上りがある場合 ($TEMP \geq 0x100$; $Cc = 1$)、桁上りががない場合には S159 で 0xFF である場合、0xFF 未満である場合に分けてバイト出力を処理する。S152 の判定が Yes の場合、すでに C レジスタ 30A から出力済みの変数 BUFFER51 値に S153 で桁上りの 1 を加えた値、S154 で変数値 SC52 個のバイト値 0 (蓄積していた 0xFF が桁上りの伝搬により 0x00 となった) の合計 SC+1 バイトが桁上りを伝搬された符号値として確定する。

【0031】S155 で変数 SC52 を 0 とし、S156 で変数 BUFFER51 に変数 TEMP の下位 8 ビッ

トを設定する。S157で変数TEMP53として処理したCレジスタ35、Cbレジスタ34をクリアし、S158で次の出力まで8ビットを処理するため変数CT50に8を設定する。S159の判定がYesの場合は、符号4は確定できず、変数値SC52に1を加えて0xFFを蓄積する。S159の判定がNoの場合は、すでにCレジスタ30Aから出力済みの符号4をS153でBUFFER51値、S154で変数値SC52個のバイト値0xFFの合計SC+1バイトが符号値として確定する。S163で変数SC52を0とし、S164で変数BUFFER51に変数TEMP53（桁上がないためそのまま8ビット）を設定する。

【0032】図52は、符号化開始時のSTテーブル8、MPSテーブル7および各変数の初期値を設定するINITENC処理を説明するフローチャートである。S171の「このレイヤの第1ストライプ」はレイヤおよびストライプの概念を持ち込まない場合「画像の符号化開始の時点」を意味し、複数のストライプから構成される画像では変数テーブルをストライプごとに初期化せずに処理を続けることもできる。S171では、このレイヤの画素の第1ストライプか、またはテーブルを強制リセットするのかを判定する。

【0033】S171の判定がYesならば、S172ですべてのコンテキストCX2に対して変数テーブルST8、MPS7を初期化する。S173はSC52を、S174はAレジスタ31値を、S175はCレジスタ30Aを、S176は変数CT50を初期化する。変数CT50の初期値11はCbレジスタ34のビット数とCsレジスタ33のビット数の和であり、11ビット処理したとき最初の符号出力を行うことになる。S171の判定がNoの場合は、S177では変数テーブルの初期化を行わず、同じレイヤの直前のストライプ終端のテーブル値を再設定する。

【0034】図53は、符号化終了時に符号化レジスタ30Aに残った値を掃き出す処理を含めた後処理を行うFLUSH処理を説明するフローチャートである。S181でCLEARBITS処理を呼び出してCレジスタ30Aに残った符号の有効桁数を最小にする。S182はFINALWRITES処理を呼び出して変数BUFFER51、SC52およびCレジスタ30Aの未確定の符号4を確定して最終的に出力する。S183で符号4の第1バイトは変数BUFFER51がCレジスタ30A値の出力に先だって（整数部として）出力されるため除去する。S184で符号4は最終有効領域内の小数座標であるから、必要であれば終端に連続するバイト0x00は除去する。

【0035】図54は、符号化終了時の符号4の有効桁数を最小とするための処理を行うCLEARBITS処理を説明するフローチャートである。これによって、符号4は終端に可能な限り0x00が連続する値となる。

S191は変数TEMP53として最終有効領域の上界値の下位2バイト（Cxレジスタ32）をクリアした値を設定している。S192では上界値の下位2バイトのクリアした値とCレジスタ30A値との大小を比較する。S192の判定がYesならば、S193で変数値TEMP53にクリアし過ぎた分を戻してCレジスタ30A値とする。S192の判定がNoならば、変数値TEMP53をCレジスタ30A値とする。

【0036】図55は、符号化終了時点で確定した符号をCレジスタ30Aに残った値まで書き出すFINALWRITES処理を説明するフローチャートである。S201でCレジスタを変数値CT50で示されるビット数だけシフトして符号出力および桁上がり判定を可能とする。S202で桁上がりの有無を判定する。S202の判定がYesならば桁上がりがあり、Noならば桁上がりはない。BYTEOUT処理を説明するフローと同様にS203、S204で桁上がりのある符号値、S207、S208で桁上がりのない符号値とするCレジスタから出力済みのSC+1バイトの符号4を確定する。S205でCbレジスタ値（1バイト）、S206でその下位1バイトを出力して符号出力が完了する。

【0037】図56は、復号処理の全体の流れを示すDECODE処理を説明するフローチャートである。国際標準勧告T.82の本処理を説明するフローチャートにあるTPおよびDPに関する処理は本発明および従来例（第1の従来例および第2の従来例）とは関係がないため省略する。まず、S211ではINITDEC処理を呼び出し、復号処理の初期化を行う。S212ではコンテキストCX2を1つずつ読み出し、S213ではDECODE処理によって画素PIX3の復号を行う。S214ではストライプ（ここではライン）が終了するまでS112およびS113を繰り返し、さらにS215では画像が終了するまでS212からS214までストライプの復号を繰り返す。

【0038】図57は、復号対象画素を復号するDECODE処理を説明するフローチャートである。まず、S221で一時的にAレジスタ31値は、下方領域幅に更新される。S222の判定がYesならば、下方領域を復号する。S223の判定がYesならば、S224でMPS_EXCHANGE処理を呼び出して、S225でRENORMD処理を呼び出して正規化処理を行う。S223の判定がNoならば、正規化処理することなく優勢シンボルを復号し、予測値7を画素値3とする。また、S222の判定がNoならば、上方領域を復号する。S227でLPS_EXCHANGE処理を呼び出して、S228でRENORMD処理を呼び出して正規化処理を行う。MPS_EXCHANGE処理、LPS_EXCHANGE処理を呼び出すパスでは、それぞれ復号すべき領域は決まっていなくても領域の大小を比較しないと復号対象シンボルが優勢シンボルか劣勢シンボルか

判断できない。呼び出した処理を説明するフローチャートでそれぞれ復号される画素値 3 を決定する。

【0039】図 58 は、上方領域を復号する LPS_EXCHANGE 処理を説明するフローチャートである。

S231 の判定が Yes ならば優勢シンボルを復号する。S232 で Cレジスタ 30B を、S233 で Aレジスタ 31 を更新する。S234 では予測値 7 をそのまま画素値 3 とする。S235 で NMPS テーブル 10 参照による状態遷移を行う。また、S231 の判定が No ならば劣勢シンボルを復号する。S236 で Cレジスタ 30B を、S237 で Aレジスタ 31 を更新する。S238 では非予測値「1-予測値」を画素値 3 とする。S239 の判定が Yes ならば、S240 で予測値 (MPS テーブル) 7 の反転・更新を行う。S241 で NLPS テーブル 11 参照による状態遷移を行う。

【0040】図 59 は、下方領域を復号する MPS_EXCHANGE 処理を説明するフローチャートである。

S251 の判定が Yes ならば劣勢シンボルを復号する。S252 では非予測値を画素値 3 とする。S253 の判定が Yes ならば、S254 で予測値 (MPS テーブル) 7 の反転・更新を行う。S255 で NLPS テーブル 11 参照による状態遷移を行う。また、S251 の判定が No ならば優勢シンボルを復号する。S256 では予測値 7 をそのまま画素値 3 とする。S257 で NMPS テーブル 10 参照による状態遷移を行う。

【0041】図 60 は、正規化処理を行う RENORM D 処理を説明するフローチャートである。S261 で変数 CT50 が 0 か否か判定し、判定が Yes ならば S262 で BYTE IN 処理を呼び出して Cレジスタ 30B に 1 バイト符号 4 を入力する。S263 では Aレジスタ 31、S264 では Cレジスタ 30B を 1 ビット上位へシフトすることによって 2 倍の乗算と等価な演算を行う。S265 で変数 CT50 から 1 を減じる。S266 の判定は、正規化処理の終了判定を行っており、Aレジスタ 31 が 0x8000 未満ならば S261 から S265 を繰り返す。S267 で変数 CT50 値が 0 か否か判定し、判定が Yes ならば S268 で BYTE IN 処理を呼び出して Cレジスタ 30B に 1 バイト符号を入力する。

【0042】図 61 は、Cレジスタ 30B に符号 4 を 1 バイトずつ読み込む BYTE IN 処理を説明するフローチャートである。SCD (Stripe Coded Data) はストライプに対する符号 4 である。S271 で判定が Yes の場合、読み出す符号 4 がいないため変数 BUFFER51 は 0 とする。S273 で変数 BUFFER51 を CLOW レジスタ 36 (Cb37) に読み込み、S274 で次の入力まで 8 ビットを処理するため変数 CT50 に 8 を設定する。また、判定が No の場合、1 バイト符号 4 を変数 BUFFER51 に読み込む。

【0043】図 62 は、復号開始時の ST テーブル 8、

MPS テーブル 7 および各変数の初期値を設定する INITDEC 処理を説明するフローチャートである。テーブルの初期値設定に関する S281、S282、S290 については、符号化処理の INITENC 処理を説明するフローチャートの S171、S172、S177 と同様である。Cレジスタ 30B の初期値は、Cx レジスタ 39 および Cb レジスタ 37 に符号 4 を 3 バイト読み込むことで設定される。S283 で Cレジスタ 30B をクリアし、S284 で BYTE IN 処理を呼び出して符号 4 を Cb レジスタ 37 に 1 バイト読み込む。S285 で Cレジスタ 30B を 8 ビットシフトして、S286 で BYTE IN 処理を呼び出して符号 4 を Cb レジスタ 37 に 1 バイト読み込む。S287 で Cレジスタ 30B を 8 ビットシフトして、S288 で BYTE IN 処理を呼び出して符号 4 を Cb レジスタ 37 に 1 バイト読み込む。これで、合計 3 バイトの符号 4 を Cx レジスタ 39 および Cb レジスタ 37 に設定したことになる。S289 では Aレジスタ 31 の初期値を設定する。

【0044】第 2 の従来例について

次に第 2 の従来例について説明する。第 2 の従来例による算術符号化の符号器・復号器は、日本国特許第 2755091 号に記載されるテーブルおよび処理を説明するフローチャートによって実現されるものである。ここで、符号をバイト単位に出力する算術符号化では、その出力バイト値が 0xFF となると、後に桁上りが発生すると出力済みのより上位まで伝播する可能性があるため符号値を確定できない。そこで、出力バイト値が 0xFF となる時点で有効領域に桁上りの可能性があれば強制的に桁上り有無を確定させてしまうことを考える。

【0045】図 63 に強制的に桁上り有無を確定させる概念図を示す。図において、下界値 C は符号レジスタ (C レジスタ) 値 30A であり、上界値 U60 は下界値 C30A に有効領域幅 A (A レジスタ値) 31 を加えた値となる。下界値および上界値の最上位ビットが一致しない場合に桁上りありと判定され、そのとき有効領域内に桁上り境界値 T61 が存在することになる。

【0046】よって、有効領域が桁上り境界によって分割される 2 つの部分領域で、桁上りがある領域を桁上り領域、桁上りがない領域を非桁上り領域と呼ぶものとすれば、桁上り領域幅 R162 は (U-T)、非桁上り領域幅 R063 は (T-C) と表される。桁上りの有無を確定するには、有効領域が桁上り領域または非桁上り領域に完全に一致するか、または含まれればよいことになり、ここでは強制的に有効領域をその部分領域の一方を切り捨て、他方を新たな有効領域として修正する。領域の切り捨てに当たり、より小さい領域を切り捨てた方が符号長のロスが少なくできるため、図中では、より小さい R063 の部分領域を切り捨てている。部分領域を切り捨てることによって、修正

された有効領域に対する正規化処理の必要性を再度判定する必要がある。このような算術符号化の桁上がり制御方法を適応領域切捨て方式と呼ぶものとする。

【0047】適応領域切捨て方式を適用した算術符号化の符号部1B、復号部16Bの概略構成を図64、図65に示す。図64では、算術符号器13B、符号4Bの置換えに伴う符号化部1Bが第1の従来例の図41に対応して構成される。図65では、算術復号器17B、符号4Bの置換えに伴う復号部16Bが第1の従来例の図42に対して異なる。これらを置き換えたブロック間のデータの流れ、また学習方法は従来形態の図41と図42と同様とする。

【0048】算術符号器13Bは、LSZ値9と2値シンボル14の入力から、Cレジスタ30AとAレジスタ31で符号化演算が進められ、CTカウンタ50でバイト単位の符号出力タイミングを計り、符号出力時に同期して領域内に桁上がり境界の有無をUレジスタ60、Tレジスタ61、R1レジスタ62、R0レジスタ63から判定し、桁上がり境界があれば適応領域切捨て方式を適用してCレジスタ30A、Aレジスタ31を修正して桁上がりの有無を強制確定し、最大BUFFER51までの桁上がり伝搬を行って確定した符号出力を行い、符号4Bを出力する。

【0049】また、算術復号器17Bは、LSZ値9と符号4Bの入力から、CTカウンタ50でバイト単位の符号入力タイミングを計り、符号入力時に同期して領域内に桁上がり境界の有無を算術符号器13Bの符号レジスタ30Aを再現したDレジスタ64、Uレジスタ60、Tレジスタ61、R1レジスタ62、R0レジスタ63から判定し、桁上がり境界があれば適応領域切捨て方式を適用してCレジスタ30A、Dレジスタ64、Aレジスタ31を修正して、BUFFER51を介して、Cレジスタ30Bと、Aレジスタ31、Dレジスタ64で復号演算が進められ、2値シンボル14を出力する。

【0050】図64と図65に示される構成ブロック内およびブロック間の詳細な処理動作は、説明済みの第1の従来例の処理を説明するフローチャートである図46から図49、図53、図54、図56、図57、図59と、後述の処理を説明するフロー図の図67から図76で説明する。

【0051】図66に適応領域切捨て方式の符号レジスタを示す。上記第1の従来例に係る図45に示した符号レジスタと比較してCsレジスタ33が削除されている。これは、符号器のバイト出力と復号器のバイト入力の同期をとるためであり、適応領域切捨て処理の適用もバイト入出力と同期して適用するものとする。ここで、符号化(符号器)では、符号レジスタ値は有効領域の下界値、復号(復号器)では、符号レジスタ値は有効領域の下界値から符号値までの変位(オフセット)に更新されていくという違いがあるため、復号時には符号器の符

号レジスタ値を再現していかないと桁上がり境界を正確に判断できない。よって、復号では、符号器の符号レジスタと同じ構成をとるDレジスタ64でその値を保持していくものとする。

【0052】本第2の従来例において、修正あるいは追加された処理を説明するフローチャートについて説明する。適応領域切捨て方式を、上記第1の従来例で説明したQM-Coderの符号化/復号処理を説明するフローチャートに必要な修正のみ加えて説明を行う。符号器に関する図46から図49、図53、図54、復号器に関する図56、図57、図59については、上記第1の従来例の処理を説明するフローと同様である。また、上記説明において使用した有効領域の上界値U60、桁上がり境界値T61、非桁上がり領域幅R063、桁上がり領域幅R162、Dレジスタ64を変数として追加導入する。

【0053】図67は、正規化処理を行うRENORME処理を説明するフローチャートである。上記第1の従来例(図50)に対して、適応領域切捨て処理の適用を判定するためのROUNDDOWNE処理を呼び出すS147が追加され、BYTEOUT処理の呼び出し(図50のS145)はROUNDDOWNE処理から行われる。

【0054】図68は、適応領域切捨て処理を適用するROUNDDOWNE処理を説明するフローチャートである。S301では出力バイト値が0xFFであるかを判定し、Noならばそのまま終了する。S302で桁上がり境界値T61(定数)を設定し、S303で非桁上がり領域幅R063を設定する。S304で非桁上がり領域幅R063を領域幅(Aレジスタ値)と比較し、R0値63がAレジスタ値31より小さくない、またはR0値63が0より大きくないならば(No)、桁上がり境界T61が有効領域内にないためそのまま終了する。

R0値63がA値31より小さい、またはR0値63が0より大きいならば(Yes)、S305で上界値U60を設定し、S306で桁上がり領域幅R162を設定する。S307でR0値63がR1値62より小さければ(Yes)、桁上がり領域を有効領域にするため、S308で下界値を桁上がり境界値に変更し、S309で領域幅(Aレジスタ値)にR1値を設定し、大きければ(No)、非桁上がり領域を有効領域にするためS309で領域幅(Aレジスタ値)31にR0値63を設定する。最後にS311でBYTEOUT処理を呼び出す。

【0055】図69(BYTEOUT処理を説明するフロー)、図70(INITENC処理を説明するフロー)及び図71(FINALWRITES処理を説明するフロー)の図51、図52及び図55に対するそれぞれの変更を説明すると、まず、符号レジスタのCsレジスタ部33の削除によって、図69(BYTEOUT処理を説明するフロー)のS151'、S157'、図70(INITENC処理を説明するフロー)のS17

6'、図71(FINALWRITES処理を説明するフロー)のS202'、S205'、S206'のマスクやシフト数の設定が変更される。

【0056】また、図68(ROUNDDOWNE処理を説明するフロー)において、出力バイト値が0xFFであるときの桁上がり伝搬の有無についてすでに解決済みであるため、変数BUFFERを逐次出力していくことができるようになり、図69(BYTEOUT処理を説明するフロー)の判定S159は不要となり、同時に変数SC自体ととそれに関わる図69(BYTEOUT

処理を説明するフロー)のS154、S155、S160、S162、S163、図70(INITENC処理を説明するフロー)のS173、図71(FINALWRITES処理を説明するフロー)のS204、S208も不要となる。

【0057】図72は、上方領域を復号するLPS_EXCHANGE処理を説明するフローチャートである。

上記第1の従来例(図58)に対して、符号器の符号レジスタをDレジスタ64として再現するためのS242、S243が追加されている。

【0058】また、図73は、正規化処理を行うRENDORMD処理を説明するフローチャートである。上記第1の従来例(図60)に対して、符号器の符号レジスタをDレジスタとして再現するためのS269、適応領域切捨て処理の適用を判定するためのROUNDDOWN処理を呼び出すS270が追加され、BYTEOUT処理の呼び出し(図60のS262)はROUNDDOWN処理から行われる。また、図60のS261とS267の判定の冗長性を除去したため処理を説明するフロー形状が異なるが、処理内容は同様である。

【0059】図74は、適応領域切捨て処理を適用するROUNDDOWN処理を説明するフローチャートである。図68のROUNDDOWNE処理に対して、符号レジスタ変数名C30AがD64に変更されており、S321からS331はS301からS311に対応し、復号器の符号レジスタ30Bに符号器の符号レジスタ30A(レジスタD64)に対応する適応領域切捨て処理を行うためのS332が追加されている。また、最後に図68のBYTEOUT処理(S311)の代わりにS331でBYTEIN処理を呼び出す。

【0060】図75(BYTEIN処理を説明するフロー)、図76(INITDEC処理を説明するフロー)の図61、図62に対するそれぞれの変更として、符号器の符号レジスタ30AをDレジスタ64として再現するための初期化処理である図75のS276、図76のS290が追加されている。

【0061】

【発明が解決しようとする課題】ところで、上述した従来例では次のような問題点がある。第1の従来例に係る符号器では、桁上がりの有無が解決せずに符号値が確定

できない場合にカウンタで待機する符号長を記憶させるが、データ長の制限がないため桁あふれする可能性があるという課題があった。また、符号値が確定しなければ最後まで符号が掃き出されないこともあり、特にデータ長が不明な場合には符号の確定遅延時間を見積もれないという課題があった。さらに符号値が確定したとき、出力を待機していた符号長が長ければその掃き出し時間も長くなり、符号化処理を全くまたは一時的に停止させなければならないという課題があった。

【0062】他方、第2の従来例に係る符号器では、符号レジスタの値により桁上がり制御の実行を判定し、桁上がり境界を厳密に計算しなければならないという課題があった。また、第2の従来例に係る復号器では、符号器の符号レジスタを再現しなければ桁上がり制御の正確な実行判定ができないという課題があった。

【0063】この発明は、以上のような従来例に係る課題を解決するためになされたものであり、符号が確定するまで待機する符号長を記憶するカウンタの桁あふれをなくし、適当な間隔で符号を強制的に確定することにより桁上がりを制御して符号の確定遅延時間を有限化して見積もれるようにすると共に、符号値が確定したときに所定のパターンへの置き換えによって総符号長と符号の掃出し時間を短縮することを目的とする。

【0064】また、桁上がり境界を厳密に計算せずに桁上がり制御を行い、復号器で符号器の符号レジスタを再現せずに桁上がり制御を行うことを目的とする。

【0065】

【課題を解決するための手段】この発明に係る符号化装置は、情報源データを蓄積して符号化対象データとその補助的なパラメータ(コンテキスト)を出力するデータメモリと、上記補助パラメータを基に指定される上記符号化対象データに関する学習データを蓄積し出力する学習メモリと、上記学習データを基に指定される符号化パラメータを出力する確率推定テーブルと、上記符号化対象データと上記符号化パラメータを基に算術符号化を行い符号を出力する符号器とを備える符号化装置において、所定の単位の情報源データの入力処理または符号の出力を所定の間隔として計測して通知する同期検出器と、上記所定の間隔で有効領域内の桁上がり境界値を検出すると共に、その検出結果に基づいて有効領域内の切り捨て領域を指示する桁上がり境界検出器とを備え、上記符号器は、上記桁上がり検出器により指示された上記有効領域を上位と下位に等分した部分領域の一方を切り捨てて上記有効領域を更新することを特徴とするものである。

【0066】また、上記符号器は、上記桁上がり境界検出器で桁上がり境界が検出されたとき、上記桁上がり境界検出器により指示された上記有効領域を上位と下位に等分した部分領域の一方を切り捨てて上記有効領域を更新することを特徴とするものである。

【0067】また、上記桁上がり境界検出器は、検出された桁上がり境界を含む部分領域を切り捨てるように指示することを特徴とするものである。

【0068】また、上記所定の間隔は、所定単位の符号化対象データを符号化したときであることを特徴とするものである。

【0069】また、上記所定の間隔は、符号値の確定まで待機させる符号長の最大値を基準に設定されることを特徴とするものである。

【0070】また、上記符号器は、上記所定の間隔を基準に符号値の確定まで待機させる符号長の最大値が設定されることを特徴とするものである。

【0071】また、上記所定の間隔は、所定単位の符号出力が発生したときであることを特徴とするものである。

【0072】また、上記所定の間隔は、所定単位の符号出力が発生し、かつ符号出力値が所定値となるときであることを特徴とするものである。

【0073】また、上記桁上がり境界検出器は、上記有効領域内に桁上がり境界が検出されなくても上記有効領域を上位と下位に等分した部分領域のどちらか一方の部分領域を指示することを特徴とするものである。

【0074】また、上記符号器は、確定した符号に同じ値が連続するときにその値と長さを特定できるランレングスマーカに置き換えるランレングスマーカ変換器を有することを特徴とするものである。

【0075】また、上記ランレングスマーカ変換器は、置き換えずにそのまま出力した方が変換される上記ランレングスマーカより符号出力が短いときには変換を行わないことを特徴とするものである。

【0076】また、上記ランレングスマーカから抽出される連続長の最大値は、復号の際のランレングスマーカ逆変換の結果に基づいて上記符号器が上記ランレングスマーカ変換器に設定することを特徴とするものである。

【0077】また、この発明に係る復号化装置は、復号対象データに対する補助的なパラメータ（コンテキスト）を出力し復号された上記復号対象データを蓄積して情報源データとして出力するデータメモリと、上記補助パラメータを基に指定される上記復号対象データに関する学習データを蓄積し出力する学習メモリと、上記学習データを基に指定される復号パラメータを出力する確率推定テーブルと、上記復号パラメータと符号を基に算術復号を行い上記復号対象データを出力する復号器とを備える復号化装置において、所定の単位の情報源データの出力処理または符号の入力を所定の間隔として計測して通知する同期検出器と、上記所定の間隔で有効領域内の桁上がり境界値を検出すると共に、その検出結果に基づいて有効領域内の切り捨て領域を指示する桁上がり境界検出器とを備え、上記復号器は、上記有効領域を上位と下位に等分した部分領域の上記符号値を含まない一方を

切り捨てて上記有効領域を更新することを特徴とするものである。

【0078】また、上記復号器は、上記桁上がり境界検出器で桁上がり境界が検出されたとき、上記有効領域を上位と下位に等分した部分領域の上記符号値を含まない一方を切り捨てて上記有効領域を更新することを特徴とするものである。

【0079】また、上記桁上がり境界検出器は、検出された桁上がり境界を含む部分領域を切り捨てるように指示することを特徴とするものである。

【0080】また、上記所定の間隔は、所定単位の復号対象データを復号したときであることを特徴とするものである。

【0081】また、上記所定の間隔は、所定単位の符号入力が発生したときであることを特徴とするものである。

【0082】また、上記所定の間隔は、所定単位の符号入力が発生し、かつ復号と同時に再現される符号出力値が所定値となるときであることを特徴とするものである。

【0083】また、上記復号器は、ランレングスマーカを検出すると、置き換えられた長さの元の符号値に逆変換するランレングスマーカ逆変換器を有することを特徴とするものである。

【0084】また、この発明に係る符号化復号化装置は、符号化装置として、情報源データを蓄積して符号化対象データとその補助的なパラメータ（コンテキスト）を出力するデータメモリと、上記補助パラメータを基に指定される上記符号化対象データに関する学習データを蓄積し出力する学習メモリと、上記学習データを基に指定される符号化パラメータを出力する確率推定テーブルと、上記符号化対象データと上記符号化パラメータを基に算術符号化を行い符号を出力する符号器と、所定の単位の情報源データの出力処理または符号の出力を所定の間隔として計測して通知する同期検出器と、上記所定の間隔で有効領域内の桁上がり境界値を検出すると共に、その検出結果に基づいて有効領域内の切り捨て領域を指示する桁上がり境界検出器とを備え、上記符号器は、上記桁上がり検出器により指示された上記有効領域を上位と下位に等分した部分領域の一方を切り捨てて上記有効領域を更新すると共に、復号化装置として、復号対象データに対する補助的なパラメータ（コンテキスト）を出力し復号された上記復号対象データを蓄積して情報源データとして出力するデータメモリと、上記補助パラメータを基に指定される上記復号対象データに関する学習データを蓄積し出力する学習メモリと、上記学習データを基に指定される復号パラメータを出力する確率推定テーブルと、上記復号パラメータと符号を基に算術復号を行い上記復号対象データを出力する復号器と、所定の単位の情報源データの出力処理または符号の出力を所定の間

隔として計測して通知する同期検出器と、上記所定の間隔で有効領域内の桁上がり境界値を検出すると共に、その検出結果に基づいて有効領域内の切り捨て領域を指示する桁上がり境界検出器とを備え、上記復号器は、上記有効領域を上位と下位に等分した部分領域の上記符号値を含まない一方を切り捨てて上記有効領域を更新することを特徴とするものである。

【0085】また、上記符号器は、上記桁上がり境界検出器で桁上がり境界が検出されたとき、上記桁上がり境界検出器により指示された上記有効領域を上位と下位に等分した部分領域の一方を切り捨てて上記有効領域を更新すると共に、上記復号器は、上記桁上がり境界検出器で桁上がり境界が検出されたとき、上記有効領域を上位と下位に等分した部分領域の上記符号値を含まない一方を切り捨てて上記有効領域を更新することを特徴とするものである。

【0086】また、この発明に係る符号化方法は、

(a) 情報源データを蓄積して符号化対象データとその補助的なパラメータ（コンテキスト）を出力するデータ蓄積ステップと、(b) 上記補助パラメータを基に指定される上記符号化対象データに関する学習データを蓄積し出力する学習ステップと、(c) 上記学習データを基に指定される符号化パラメータを出力する確率推定ステップと、(d) 上記符号化対象データと上記符号化パラメータを基に算術符号化を行い符号を出力する符号化ステップと、(e) 所定の単位の情報源データの出力処理または符号の出力を所定の間隔として計測して通知する同期検出ステップと、(f) 上記所定の間隔で有効領域内の桁上がり境界値を検出すると共に、その検出結果に基づいて有効領域内の切り捨て領域を指示する桁上がり境界検出ステップと、(g) 上記桁上がり境界検出ステップにより指示された上記有効領域を上位と下位に等分した部分領域の一方を切り捨てて上記有効領域を更新する符号化修正ステップとを有するものである。

【0087】また、上記符号化修正ステップは、上記桁上がり境界検出ステップで桁上がり境界が検出されたとき、上記桁上がり境界検出ステップで指示された上記有効領域を上位と下位に等分した部分領域の一方を切り捨てて上記有効領域を更新することを特徴とするものである。

【0088】また、上記符号化ステップおよび上記符号化修正ステップは、確定した符号に同じ値が連続するときとその値と長さを特定できるランレングスマーカに置き換える変換ステップを有することを特徴とするものである。

【0089】上記変換ステップは、置き換えずにそのまま出力した方が変換される上記ランレングスマーカより符号出力が短いときには変換を行わないことを特徴とするものである。

【0090】また、上記ランレングスマーカから抽出さ

れる連続長の最大値は、復号の際、ランレングスマーカ逆変換の結果に基づいて上記符号化ステップが上記変換ステップに設定することの特徴とするものである。

【0091】また、この発明に係る復号化方法は、

(a) 復号対象データに対する補助的なパラメータ（コンテキスト）を出力し復号された上記復号対象データを蓄積して情報源データとして出力するデータ蓄積ステップと、(b) 上記補助パラメータを基に指定される上記復号対象データに関する学習データを蓄積し出力する学習ステップと、(c) 上記学習データを基に指定される復号パラメータを出力する確率推定ステップと、(d) 上記復号パラメータと符号を基に算術復号を行い上記復号対象データを出力する復号化ステップと、(e) 所定の単位の情報源データの出力処理または符号の入力を所定の間隔として計測して通知する同期検出ステップと、

(f) 上記所定の間隔で有効領域内の桁上がり境界値を検出すると共に、その検出結果に基づいて有効領域内の切り捨て領域を指示する桁上がり境界検出ステップと、

(g) 上記桁上がり境界検出ステップにより指示された上記有効領域を上位と下位に等分した部分領域の一方を切り捨てて上記有効領域を更新する復号化修正ステップとを有するものである。

【0092】また、上記復号化修正ステップは、上記桁上がり境界検出ステップで桁上がり境界が検出されたとき、上記桁上がり境界検出ステップで指示された上記有効領域を上位と下位に等分した部分領域の一方を切り捨てて上記有効領域を更新することを特徴とするものである。

【0093】さらに、上記復号化ステップは、ランレングスマーカを検出すると、置き換えられた長さの元の符号値に逆変換するランレングスマーカ逆変換ステップを有することを特徴とするものである。

【0094】

【発明の実施の形態】実施の形態1. 本実施の形態1では、上述した第2の従来例と同様に、有効領域の部分領域を切り捨て、出力済みの符号に対して桁上がり有無を確定する方法をとるが、有効領域を2つの部分領域に等分割し、桁上がり境界を含む部分領域を切り捨てるものである。

【0095】領域幅は常に符号器と復号器とで同値をとるため、等分割操作によれば、上記第2の従来例のように桁上がり境界値を厳密に符号化／復号に持ち込まなくても上位／下位のどちらの部分領域に含むかだけわかればよいことになる。

【0096】例えば、図1に示すように、最大有効領域を16とするとき、正規化処理により有効領域は8以上に保たれる。このとき、1.0000を桁上がり境界とすれば、その境界が上位にあれば上位1/2領域を切り捨て、下位にあれば下位1/2領域を切り捨てる。ただし、有効領域幅が奇数のとき、有効な表現精度の不足に

10

20

30

40

50

よる桁落ちによる誤差が生じるため、その解決策も含める必要がある。同図では、有効領域幅＝9であるため、単純に $1/2$ とすれば4（整数）となり、それを下位領域幅とすると上位領域幅は5となる。よって、上位領域を切り捨てれば下位領域幅が2倍に拡大されて8、下位領域を切り捨てれば上位領域幅が2倍に拡大されて10となり、厳密に等しく分割されたことにならないことになる。

【0097】ここで、領域幅（Aレジスタ）31の $1/2$ を計算した後、正規化処理で少なくとも2倍に拡大されることは確実である。よって、Aレジスタの $1/2$ 演算で桁落ちするビットを後から補正する方法がある。また、もう1つの方法は、領域幅（＝9）は保ったまま、まず下界値（Cレジスタ）30Aだけを2倍し、下位領域を切り捨てるときその領域幅 $31=9$ を加えれば所望の誤差のない座標を得ることができる。いずれにしても、 $1/2$ 領域切捨て方式によれば、1回適用につき必ず1ビットの符号長が発生することになる。

【0098】本実施の形態1では、 $1/2$ 領域切捨て方式の算術符号化の符号器を符号部1C、その復号器を復号部16Cとし、その符号器、復号器の概略構成を図2、図3に示す。図2では、算術符号器13C、符号4Cの置換えに伴う符号化部1Cが第2の従来例の図64に対応して構成される。図3では、算術復号器17C、符号4Cの置換えに伴う復号部16Cが第2の従来例の図65に対して異なる。これらを置き換えたブロック間のデータの流れ、また学習方法は従来例の図64と図65と同様とする。

【0099】算術符号器13Cは、LSZ値9と2値シンボル14の入力から、Cレジスタ30AとAレジスタ31で符号化演算が進められ、CTカウンタ50でバイト単位の符号出力タイミングを計り、符号出力時に同期して領域内に桁上がり境界の有無をTレジスタ61、R0レジスタ63から判定し、桁上がり境界があれば $1/2$ 領域切捨て方式を適用してCレジスタ30A、Aレジスタ31を修正して桁上がりの有無を強制確定し、最大BUFFER51までの桁上がり伝搬を行って確定した符号出力を行い、符号4Cを出力する。

【0100】また、算術復号器17Cは、LSZ値9と符号4Cの入力から、CTカウンタ50でバイト単位の符号入力タイミングを計り、符号入力時に同期して領域内に桁上がり境界の有無を算術符号器13Cの符号レジスタ30Aを再現したDレジスタ64、Tレジスタ61、R0レジスタ63から判定し、桁上がり境界があれば $1/2$ 領域切捨て方式を適用してCレジスタ30A、Dレジスタ64、Aレジスタ31を修正して、BUFFER51を介して、Cレジスタ30Bと、Aレジスタ31、Dレジスタ64で復号演算が進められ、2値シンボル14を出力する。

【0101】図2と図3に示される構成ブロック内およ

びブロック間の詳細な処理動作は、説明済みの第1の従来例の処理を説明するフロー図である図46から図49、図53、図54、図56、図57、図59と、同じく第2の従来例の処理を説明するフローチャートである図67、図69から図74、図75、図76と、後述の処理を説明するフロー図4から図7、またはその等価な処理を説明するフローとして与えられる図4、図6、図8から図11で説明する。

【0102】本実施の形態1では、第2の従来例において桁上がりまたは非桁上がり領域領域の切捨て処理を行っている図68（ROUNDDOWNE処理を説明するフロー）、図74（ROUNDDOWNND処理を説明するフロー）を次に示す図4、図6へ変更し、そこから呼び出される図5（HALVINGE処理を説明するフロー）、図7（HALVINGD処理を説明するフロー）を追加することによって $1/2$ 領域切捨て方式を実現する方法を示す。

【0103】図4は、 $1/2$ 領域切捨て処理の適用を判定するROUNDDOWNE処理を説明するフローチャートである。図中、S301からS304、およびS301、S304の判定結果（いずれもNo）により処理されるS311は図68と同じである。S304の判定結果（Yes）によりS312で $1/2$ 領域切捨て処理を行うHALVINGE処理を呼び出す。

【0104】図5は、 $1/2$ 領域切捨て処理を行うHALVINGE処理を説明するフローチャートである。S341で非桁上がり領域幅R0の2倍がAレジスタ値31より小さければ（判定Yes）、下位 $1/2$ 領域が桁上がり境界を含み、小さくなければ（判定No）、情報 $1/2$ 領域が桁上がり領域を含むことになり、桁上がり境界を含む $1/2$ 領域を切り捨てるものとする。下位 $1/2$ 領域を切り捨てる場合、S342でCレジスタ値（有効領域下界値）30AをAレジスタ値（有効領域幅）31の $1/2$ だけ更新した後、S343でBYTEOUT処理を呼び出す。領域切捨てに伴い、実際はAレジスタ値31は $1/2$ とするべきであるが、すぐに正規化処理で2倍に拡大されるため更新していない。Cレジスタ30AとCTカウンタ値50に対して正規化処理に相当するS344、S345を行う。S346でAレジスタ値31が奇数か否かを判定し、奇数であればAレジスタ値31の $1/2$ 倍で桁落ちした分をS347で補正する。上位 $1/2$ 領域を切り捨てる場合、Cレジスタ値30Aは更新されず、S348でBYTEOUT処理を呼び出す。Cレジスタ30AとCTカウンタ値50に対して正規化処理に相当するS349、S350を行う。

【0105】図6は、 $1/2$ 領域切捨て処理の適用を判定するROUNDDOWNND処理を説明するフローチャートである。図中、S321からS324、およびS321、S324の判定結果（いずれもNo）により処理されるS332は図74と同じである。S324の判定

10

20

30

40

50

結果 (Yes) により S333 で 1/2 領域切捨て処理を行う HALVINGD 処理を呼び出す。

【0106】図7は、1/2 領域切捨て処理を行う HALVINGD 処理を説明するフローチャートである。図5の ROUNDDOWNE 処理に対して、符号レジスタ変数名 C30A が D64 に変更されており、S361 から S370 は S341 から S350 に対応し、復号器の符号レジスタ C30B に符号器の符号レジスタ 30A

(レジスタ D64) に対応する 1/2 領域切捨て処理を行うための S371、S372、S373、S374 を追加している。また、最後に図68の BYTEOUT 処理 (S311) の代わりに S331 で BYTEIN 処理を呼び出す。

【0107】ここで、図5の HALVINGE 処理を説明するフロー、図7の HALVINGD 処理を説明するフローは、呼び出される BYTEOUT 処理を説明するフロー (図69)、BYTEIN 処理を説明するフロー (図74) とともに、次に示す図8 (HALVINGE 処理を説明するフロー)、図9 (BYTEOUT7 処理を説明するフロー)、図10 (HALVINGD 処理を説明するフロー)、図11 (BYTEIN7 処理を説明するフロー) に置き換えることができる。BYTEOUT7 処理を説明するフロー、BYTEIN7 処理を説明するフローは A レジスタ値 (領域幅) 31 を 1/2 にしないで C レジスタ 30B を先に 2 倍に拡大して桁落ちを防止するため、バイトの入出力位置が BYTEOUT 処理を説明するフロー、BYTEIN 処理を説明するフローチャートの正規の位置より上位に 1 ビットだけずれている。特に、符号器では上位に 1 ビット余分に精度が保証されていないからである。

【0108】図8は、1/2 領域切捨て処理を行う HALVINGE 処理を説明するフローチャートである。S381、S382 で C レジスタ 30A、非桁上がり領域幅 R063 を 2 倍に拡大し、正規化に相当する処理を先に行う。S307 で非桁上がり領域幅 R063 が A レジスタ値 31 より小さければ (判定 Yes)、下位 1/2 領域が桁上がり境界を含み、小さくなければ (判定 No)、上位 1/2 領域が桁上がり領域を含むことになる。下位 1/2 領域を切り捨てる場合 (判定 Yes)、S384 で C レジスタ値 (有効領域下界値) 30A を A レジスタ値 (有効領域幅) 31 だけ更新する。符号バイト出力前に本来は BYTEOUT 処理の直後に行われる 2 倍の拡大 (正規化処理) を先行しているため、S385 では、BYTEOUT 処理ではなく、BYTEOUT7 処理を呼び出す。

【0109】図9は、BYTEOUT 処理の詳細な処理を説明するフローチャートである。この BYTEOUT7 処理の図69 (BYTEOUT 処理を説明するフロー) に対する変更を示す。S151'、S157' の修正は、符号出力位置が 1 ビット上位に移動していることによる。

による。S158' の修正は、図8ですでに正規化に相当する処理 (S382) が行われていることによる。

【0110】図10は、1/2 領域切捨て処理を行う HALVINGD 処理を説明するフローチャートである。

図8の ROUNDDOWNE 処理に対して、符号レジスタ変数名 C30A が D64 に変更されており、S391 から S395 は S381 から S385 に対応し、復号器の符号レジスタ C30B に符号器の符号レジスタ 30A (レジスタ D64) に対応する処理を行うための S396、S397 を追加している。また、最後に図8の BYTEOUT7 処理 (S385) の代わりに S395 で BYTEIN7 処理を呼び出す。

【0111】図11は、BYTEIN7 処理の詳細な処理を説明するフローチャートである。この BYTEIN7 処理の図74 (BYTEIN 処理を説明するフロー) に対する変更を示す。S273'、S276' の修正は、符号入力位置が 1 ビット上位に移動していることによる。S274' の修正は、図10ですでに正規化に相当する処理 (S392、S396) が行われていることによる。

【0112】上述した実施の形態1によれば、桁上がり境界が上位と下位の 1/2 領域のどちらに存在しているかの判定されればよいから、桁上がり領域幅 R162 を計算する必要がない。

【0113】実施の形態2。前述した第1の従来例に示される QM-Coder では、出力済みであっても確定していない符号を変数 BUFFER51、SC カウンタ 52 で待機させているが、符号器の仕様によっては符号化中に SC カウンタ 52 の桁あふれが発生することがあり、結果として桁あふれ 1 回ごとにそのカウント最大数だけ符号バイト (0xFF または 0x00) が欠落し、正確な符号を出力できないことになる。

【0114】一般に、最終的な画像サイズは事前にわかっているとは限らないが、有限のサイズで区切ったストライプ (またはライン) 単位で処理を行えばその最悪の符号量を見積もることができる。算術符号化では最悪の場合 1 ビットを若干超える程度となることが知られているため、その区切ったストライプ (またはライン) 構成サイズ (バイト数) に対してはほぼ用意すべき SC カウンタ 52 の仕様が確定でき、その 2 倍の符号バイト 0xFF のランを計数できれば十分である。本実施の形態2では、ストライプ終了ごとにそこまで出力を待機している符号バイトを強制的に確定して掃き出していくことによって、最大確定遅延もストライプの符号化処理時間程度に見積もれるようになる。この符号バイトの確定手法として 1/2 領域切捨て方式を適用するものとする。

【0115】また、前述した第2の従来例に示される適応領域切捨て方式では、1 バイトの符号バイトのみ待機させれば出力を逐次行っていくことが可能であるが、領域切捨て処理による最小の符号長ロスで抑えるために有

効領域内の桁上がり境界の位置を厳密に計算しなければならないこと、また、そのため、復号器では符号器の符号レジスタ 30 A の再現を行わなければ正確な復号ができない。さらに、適応領域切捨て処理の実行を C s レジスタ 33 の削除によって符号器、復号器のバイト入出力のタイミングで同期させなければ他の専用同期手段を用意する必要があり、その判定などによって符号化／復号の負荷が増加する。例えば、C s レジスタ 33 を削除せずに符号器と復号器を同期させるために、符号器には上記専用同期手段として正規化シフト数のカウンタを設

【0116】本実施の形態 2 では、ストライプの終了のタイミングで 1/2 領域切捨て処理を行うことによって C s レジスタ 33 の有無による同期上の制約を解消できる。また、C レジスタ 30 A から出力される符号バイト値に依存せず、かつ有効領域内に桁上がり境界がないときにも必ず 1/2 領域切捨て処理を行うことによって、復号器では符号器の符号レジスタ 30 A の再現することなく、また、桁上がり境界を厳密に計算せずに符号 (C レジスタ値) 30 B が指し示さない上位または下位の 1/2 領域を切り捨てることによって正確な復号を行うことができる。

【0117】ここで、符号器は、上記のように切り捨てるべき 1/2 領域を決定し、有効領域を更新するが、これはダミーシンボルの符号化とみなせる。また、復号器は、符号の指し示す 1/2 領域に有効領域を更新するが、これはダミーシンボルの復号とみなすことができ、有効領域が上位／下位のどちらに更新されてもシンボルは無視してよい。

【0118】本実施の形態 2 の 1/2 領域切捨て方式では、ストライプの符号化処理で符号ビットが発生しなくても 1/2 領域切捨てによって 1 ビットの符号が発生する。このため、S C カウンタ 52 の桁あふれを抑えながら、最悪 8 ストライプで 1 バイトの符号を出力させられることになり、その所要時間を符号確定までの最大遅延として見積もることができる。

【0119】1/2 領域切捨て方式の算術符号化の符号器を符号部 1 C、その復号器を復号部 16 C とし、その符号器、復号器の概略構成を、図 12、図 13 に示す。図 12 では、算術符号器 13 D、符号 4 D の置換えに伴う符号化部 1 D が第 1 の従来例の図 39 に対応して構成される。図 13 では、算術復号器 17 D、符号 4 D の置換えに伴う復号部 16 D が第 1 の従来例の図 40 に対して異なる。これらを置き換えたブロック間のデータの流れ、また、学習方法は第 1 の従来例の図 39 と図 40 と同様とする。

【0120】算術符号器 13 D は、L S Z 値 9 と 2 値シンボル 14 の入力から、C レジスタ 30 A と A レジスタ

31 で符号化演算が進められ、C T カウンタ 50 でバイト単位の符号出力タイミングを計り、第 1 の従来例と同様に B U F F E R 51 と S C カウンタで桁上がりの伝搬可能性のある未確定の符号出力待機と確定した符号出力を行い、符号 4 D を出力していく。そして、ライン終端 (あるいはストライプ終端) の検出に同期して領域内に桁上がり境界の有無を T レジスタ 65、C W レジスタ 66 から判定し、選択された上位または下位の 1/2 切捨て領域を切り捨てる 1/2 領域切捨て方式を適用し、C レジスタ 30 A、A レジスタ 31 を修正して桁上がりの有無を強制確定し、最大 B U F F E R 51 までの桁上がり伝搬を行って確定した符号出力を行い、符号 4 D を出力する。

【0121】また、算術復号器 17 D は、L S Z 値 9 と符号 4 D の入力から、C T カウンタ 50 でバイト単位の符号入力タイミングを計り、B U F F E R 51 を介して、C レジスタ 30 B と、A レジスタ 31 で復号演算が進められ、2 値シンボル 14 を出力する。そして、ライン終端 (あるいはストライプ終端) の検出に同期して C レジスタ 30 B の指し示さない側の 1/2 切捨て領域を切り捨てる 1/2 領域切捨て方式を適用し、C レジスタ 30 B、A レジスタ 31 を修正する。

【0122】図 12 と図 13 に示される構成ブロック内およびブロック間の詳細な処理動作は、説明済みの第 1 の従来例の処理を説明するフローチャートである図 47 から図 55、図 57 から図 62 と、後述の処理を説明するフロー図 14 から図 19 で説明する。本実施の形態 2 では、図 14 および図 18 により、ストライプの終了時に 1/2 領域切捨て方式を適用する場合を示す。

【0123】本実施の形態 2 における 1/2 領域切捨て方式の一例を、上記第 1 の従来例で説明した Q M - C o d e r の符号化／復号処理を説明するフローチャートに必要な修正のみ加えて説明を行う。符号化／復号／領域幅レジスタに関する図 45、符号器に関する図 47 から図 55、復号器に関する図 57 から図 62 については、上記第 1 の従来例の処理を説明するフローと同様である。また、説明上、上記第 1 の従来例に追加導入される変数は、有効領域の 1/2 領域境界値 T 65、変数 C T 値 50 によって定義される桁上がり境界値 C W 66 である。

【0124】図 14 は、符号化処理の全体の流れを示す E N C O D E R 処理を説明するフローチャートである。

図 46 におけるストライプの終了判定 (S104) と画像の終了判定 (S105) の間に 1/2 領域切捨て処理を行う H A L V I N G E 処理の呼出し (S107) を追加する。

【0125】図 15 は、1/2 領域切捨て処理を行う H A L V I N G E 処理を説明するフローチャートである。S501、S502 は、C レジスタ 30 A、C T カウンタ 50 に対する正規化処理に相当し、A レジスタ 31 は

10

20

30

40

50

そのまま保つ。S503、S504は、有効領域の1/2領域境界値T65、CTカウンタ値50で定義される桁上がり境界値CW66を設定する。S505で、1/2領域境界値T65と桁上がり境界値CW66を比較判定する。判定結果Yesならば桁上がり境界は上位1/2領域以上に存在するため、桁上がり伝搬がない待機中の符号バイトを出力し、下位1/2領域を有効領域とするCODELOWER処理を呼び出す(S506)。判定結果Noならば桁上がり境界は下位1/2領域以下に存在するため、桁上がり伝搬させた待機中の符号バイトを出力し、上位1/2領域を有効領域とするCODEUPPER処理を呼び出す(S507)。最後に、S508で変数CT50が0ならば、S509でBUFFER51を書き出し、S510でCbレジスタ値34をBUFFER51に設定し、S511でCcレジスタ部35およびCbレジスタ部34をクリアし、S512でCTカウンタ50に8を設定する。ここで、すでにS506あるいはS507によって、桁上がり有無を確定し、桁上がりがあれば伝搬処理もされ(伝搬後、Ccレジスタ部35=0)、待機していた符号バイトの掃出しが完了(掃出し後、変数SC値52=0)しており、S509からS512はBYTEOUT処理(図51)において冗長部を最大限省いたS161、S164(TEMP=Cbレジスタ値)、S157、S158に相当する。S510ではBUFFER51がバイト0xFFになっても問題ない。

【0126】図16は、上位1/2領域を切り捨て、桁上がりなしの出力待機符号の掃出し処理を行うCODELOWER処理を説明するフローチャートである。S521で変数SC値52が正ならば、S522でBUFFER51を書き出す。さらに、S523で変数SC値52が1より大きければ、S524で符号バイト0xFFを(SC-1)回書き出す。S525で最後の待機符号バイトである0xFFをBUFFER51に設定し、S526で変数SC値52を0とする。

【0127】図17は、下位1/2領域を切り捨て、桁上がりありの出力待機符号の掃出し処理を行うCODEUPPER処理を説明するフローチャートである。S531でBUFFER51に桁上りを伝搬させ、Cレジスタ30Aから桁上がり分(桁上がり境界値CW66)をクリアする。S533で変数SC値52が正ならば、S534でBUFFER51を書き出す。さらに、S535で変数SC値52が1より大きければ、S536で符号バイト0x00を(SC-1)回書き出す。S537で最後の待機符号バイトである0x00をBUFFER51に設定し、S537で変数SC値52を0とする。

【0128】図18は、復号処理の全体の流れを示すDECODER処理を説明するフローチャートである。図56におけるストライプの終了判定(S214)と画像

の終了判定(S215)の間に1/2領域切捨て処理を行うHALVINGD処理の呼出し(S216)を追加する。

【0129】図19は、1/2領域切捨て処理を行うHALVINGD処理を説明するフローチャートである。

S541、S542は、Cレジスタ30B、CTカウンタ50に対する正規化処理に相当し、Aレジスタ31はそのまま保つ。S545でHIGHレジスタ値38とAレジスタ値31を比較し、判定結果Yesであれば下位1/2領域が切り捨てられたことになってS546でHIGHレジスタ値38からAレジスタ値31を減じる。判定結果Noであれば下位1/2領域が切り捨てられたことになる。最後に、S547で変数CT50が0ならば、BYTEIN処理を呼び出す。

【0130】ただし、桁上がり境界値CW66が1/2領域境界T65と等しい場合は、上位、下位のどちらの1/2部分領域をとってもよいが、例えば国際標準勧告T.82では符号バイト0xFFの直後にはSTUFFバイト(0x00)を挿入して伝送上の制御(マーカコードの確保)を行うため、桁上りを伝搬させた方が最終的な符号長が短くなることがある。

【0131】ここでは、画像を区切ったストライプ(またはライン)構成サイズ(バイト数)に対して用意すべきSCカウンタ52の仕様が確定できるとしたが、逆にSCカウンタ52の仕様が変更できないときには桁あふれしえないストライプ(またはライン)構成サイズを設定すればよい。

【0132】実施の形態3。上記実施の形態2の最大遅延が8ストライプであったが、符号ビットが発生しないときは桁上りの有無は解決されたままになっているため、1/2領域切捨て処理を行わなければ符号確定までの遅延は大きくなるが、SCカウンタ52の桁あふれを抑えながら符号中に1/2領域切捨て処理に伴う余分な符号ビットを生じさせないことも可能である。本実施の形態3では、一例として上記実施の形態2で説明したENCODER処理を説明するフロー(図14)、DECODER処理を説明するフロー(図18)を図20、図21に置き換えたものとする。

【0133】図20は、符号化処理の全体の流れを示すENCODER処理を説明するフローチャートである。

図14におけるストライプの終了判定(S104)と1/2領域切捨て処理を行うHALVINGE処理の呼出し(S107)の間にストライプの符号化処理中に正規化処理が行われた(符号ビットが発生したか)どうかの判定(S108)を追加し、行われた場合のみ1/2領域切捨て処理を行う。

【0134】図21は、復号処理の全体の流れを示すDECODER処理を説明するフローチャートである。図18におけるストライプの終了判定(S214)と1/2領域切捨て処理を行うHALVINGD処理の呼出し

【S216】の間にストライプの復号処理中に正規化処理が行われた（符号ビットが読み出された）かどうかの判定（S217）を追加し、行われた場合のみ1/2領域切捨て処理を行う。

【0135】ここで、符号レジスタC30Aの構成仕様を上記実施の形態1と同様にすれば、符号器／復号器のバイト入出力が同期可能なため、本実施の形態3において正規化処理の有無による1/2領域切捨て処理実行をライン処理中の符号バイト入出力の有無（実行有無フラグを導入または符号ビット数の増加から検出するなど）で行うことも可能である。この場合、バイト入出力までの数ビット未満の正規化処理が発生しても1/2領域切捨て処理の実行が行われないことになり、それに伴う余分な符号ビットを削減できる。

【0136】実施の形態4。上記実施の形態2では、画像をストライプに分割して符号化／復号する際に、1回のみ符号器／復号器の初期化処理および後処理を行うものとして示した。本実施の形態4では、1/2領域切捨て処理をラインの処理ごとに実行し、ストライプの処理ごとに符号器／復号器の初期化処理および後処理を行うものとする。ここで、符号器の後処理（FLUSH処理を説明するフロー）には、未確定の符号を確定して掃き出す処理が行われるため、ストライプは複数ラインから構成されるものとして、ライン終端ごとに1/2領域切捨て処理を行うものとする。本実施の形態4では、一例として上記実施の形態2で説明したENCODER処理を説明するフロー（図14）、DECODER処理を説明するフロー（図18）を図22、図23に置き換えたものとする。

【0137】図22は、符号化処理の全体の流れを示すENCODER処理を説明するフローチャートである。

図14に対してラインの終了判定（S109）を追加したことにより、ストライプの終了判定（S104）、画像の終了判定（S105）を再構成することによって、ラインごとに1/2領域切捨て処理（S107）を実行し、ストライプごとに初期化処理（S101）、後処理（S106）を実行している。

【0138】図23は、復号処理の全体の流れを示すDECODER処理を説明するフローチャートである。図18に対してラインの終了判定（S218）を追加したことにより、ストライプの終了判定（S214）、画像の終了判定（S215）を再構成することによって、ラインごとに1/2領域切捨て処理（S216）を実行し、ストライプごとに初期化処理（S211）を実行している。

【0139】このように、ラインごとに1/2領域切捨て処理を行うことによって、上記実施の形態2と同様に符号の確定遅延は最大8ラインとなり、かつ初期化処理、後処理を行うことによって符号がストライプごとに完結することになる。

【0140】実施の形態5。上記実施の形態3では、画像をストライプに分割して符号化／復号する際に、1回のみ符号器／復号器の初期化処理および後処理を行うものとして示した。本実施の形態5では、1/2領域切捨て処理をラインの処理ごとに実行し、ストライプの処理ごとに符号器／復号器の初期化処理および後処理を行うものとする。本実施の形態5では、一例として上記実施の形態2で説明したENCODER処理を説明するフロー（図20）、DECODER処理を説明するフロー（図21）を、図24、図25に置き換えたものとする。

【0141】図24は、符号化処理の全体の流れを示すENCODER処理を説明するフローチャートである。

図20に対してラインの終了判定（S109）を追加したことにより、ストライプの終了判定（S104）、画像の終了判定（S105）を再構成することによって、ラインごとに1/2領域切捨て処理（S107）を実行し、ストライプごとに初期化処理（S101）、後処理（S106）を実行している。

【0142】図25は、復号処理の全体の流れを示すDECODER処理を説明するフローチャートである。図21に対してラインの終了判定（S218）を追加したことにより、ストライプの終了判定（S214）、画像の終了判定（S215）を再構成することによって、ラインごとに1/2領域切捨て処理（S216）を実行し、ストライプごとに初期化処理（S211）を実行している。

【0143】このように、ラインごとに1/2領域切捨て処理を行うことによって、上記実施の形態3と同様に符号の確定遅延は上記実施の形態4より大きくなるが、SCカウンタ52の桁あふれを抑えながら符号中に余分な符号ビットを生じさせないことになる。また、上記実施の形態4と同様に初期化処理、後処理を行うことによって符号がストライプごとに完結することになる。

【0144】上記実施の形態2から上記実施の形態5は、符号器と復号器における1/2領域切捨て処理の同期をストライプ、ラインといった単位で行っているが、例えばブロック、サブブロックといった他の何らかの同期のとれる単位であっても構わない。また、データは必ずしも画像でなくとも構わない。

【0145】実施の形態6。本実施の形態6では、例えばバイト単位で符号を出力する場合に、符号中に同じバイト値が連続するときそのバイト値の連続を、バイト値およびその長さを意味する他のバイト群で置き換えることによって総符号長とその出力時間を短縮する方法を示す。以下、ここで置き換えるバイト群をランレングスマーカと呼び、所定の指示方法によって任意の長さに拡張して数値を表現することができるものとする。

【0146】図26に示すように、例えば4つのバイト0xFFの連続を、（1）ではエスケープコードESC

とそれに続く $0xFF$ が 4 つ続くことを示すバイト MK で置き換えることにより、ESC コードを確保するために挿入された STUFF バイトと併せて 8 バイトを 2 バイトとする例である。(2) ではバイト MK は $0xFF$ の置き換えであることのみ示し、その長さをバイト RL の 3 バイトに置き換える例である。また、 $0x00$ が続く場合も同様に $0x00$ が続くことを示すバイト MK と別途必要ならばバイト RL で置き換える例である。ここで、バイト MK はその置き換える対象のバイト値およびその長さを識別できる値が割り振られているものとする。

【0147】また、図 27 に示すように、7 個のバイト $0xFF$ を置き換えたランレングスマーカは、一例として、4 個と 3 個の置き換えを示す 2 つのランレングスマーカ、または 2 個と 3 個と 2 個の置き換えを示す 2 つのランレングスマーカとも等価であり、また他のラン長の合計が等しくなる組合せであっても等価であるといえる。

【0148】ランレングスマーカは、例えば図 28 に示すように、ESC コード ($0xFF$) とそれに続くバイト MK で示すとすれば、バイト MK は $(8-N)$ ビットの置き換え対象バイトの識別部と N ビットのランレングス部から構成される。この場合、置き換え対象バイトは 2 の $(8-N)$ 乗種類を指定でき、また 2 の N 乗までの置き換え対象バイトのラン長を示すことができる。

【0149】ただし、国際標準勧告 T. 82 では他のマーカセグメントが定義されているため、識別部で指定可能なすべての値を使用できるわけではない。また、他の国際標準勧告でも別途マーカセグメントが使用されている可能性があるため、ここでランレングスマーカを使用する際にはそれらのすべてのマーカセグメントと重複しないようにしなければならない。

【0150】また、ランレングスマーカは、例えば図 29 に示すように、ESC コード ($0xFF$) とそれに続くバイト MK、および所定の S バイトのランレングス部 RL で示すとすれば、バイト MK は 8 ビットすべてが置き換え対象バイトの識別部となる。この場合、置き換え対象バイトは 255 種類 ($MK=STUFF$ を除く) を指定でき、また 2 の $(8 \times S)$ 乗までの置き換え対象バイトのラン長を示すことができる。

【0151】さらに、ランレングスマーカは、例えば図 30 に示すように、ESC コード ($0xFF$) とそれに続くバイト MK、および S バイトのランレングス部 RL で示すとすれば、バイト MK は $(8-N)$ ビットの置き換え対象バイトの識別部と N ビットのランレングス指示部から構成される。この場合、置き換え対象バイトは 2 の $(8-N)$ 乗種類を指定でき、また 2 の N 乗バイトまでのランレングス部のバイト長を示すことができる。そして、2 の $(8 \times S)$ 乗までの置き換え対象バイトのラン長を示すことができる。

【0152】図 28 から図 30 にランレングスマーカの

一例を示したが、バイト MK で置き換え対象バイトを指示できないとき、ESC およびバイト MK でランレングスマーカの開始を判定させ、固定長またはバイト MK で指定した長さのランレングス部に、さらに置き換え対象バイトをランレングス部の前または後に置くことで直接指定する方法もある。この方法によれば、他のマーカセグメントに対して、ランレングスマーカであることだけをバイト MK で認識させれば、置き換え対象バイトはその値によらずすべてが使用できるようになる。

【0153】また、複数バイトの繰返しに対しても、例えばバイト MK にその繰返される置き換え対象パターンのバイト長を示し、パターンのラン長 (繰返し数) とともにパターンを直接指定する方法もある。

【0154】一例を図 31 (a) に示すと、ESC コード ($0xFF$) と識別部を含むマーカ MK、複数バイトから構成される置き換え対象パターン部 PT、ランレングス部 RL から構成されるものとする。ここで、マーカ MK は、置き換え対象パターン部 PT のフィールド長 (値 P)、ランレングス部 RL のフィールド長 (値 R) を指定するとすれば P バイトの置き換え対象パターン PT が R バイトで表現される RL 回繰返されることになる。

【0155】その具体例として、3 回繰返される 2 バイトのパターン $0x01$, $0x23$ をランレングスマーカに置き換えると、置き換え対象パターン部のフィールド長 P が 2、ランレングス部のフィールド長 R が 1 であるマーカ MK を使用して、図 31 (b) のように示される。

【0156】マーカ MK において、ランレングスマーカ識別部を除く上記 2 つのフィールド長部は、その識別部を含むバイトと独立であっても構わない。また、置き換え対象パターン部 PT のフィールド長部とランレングス部 RL のフィールド長部、さらに置き換え対象パターン部 PT とランレングス部 RL は、符号器と復号器の間で認識が一致すれば上記一例に限らずそれぞれ自由に配置することができる。

【0157】ランレングスマーカに指示される置き換え対象バイトのラン長は 0 であることはないため、0 が最大値を表すか、または実際の (ラン長-1) を常にランレングス部に埋め込むかを予め決めておく必要がある。

【0158】ここで、符号器における $0xFF$ または $0x00$ の連続した SC 回のバイト出力 (BYTEOUT 処理を説明するフローチャートの S154、S162、FINALWRITES 処理を説明するフローチャートの S204、S208、CODELOWER 処理を説明するフローチャートの S524、CODEUPPER 処理を説明するフローチャートの S536) については、例えば図 32 に示すように、RUNLENNMARK 処理を説明するフローチャートでランレングスマーカへの置き換え処理を容易に実行することができる。ここで、符号器と復号器では、上記図 28 から図 30 のどの形式のランレングスマーカを使用するかということを予め取り決

10

20

30

40

50

めているものとする。その使用するランレングスマーカのランレングス部に示すことのできる最大値を定数MARKLEN80とする。例えば、図28において、N=4であれば、定数MARKLEN80は16となる。

【0159】図32において、RL81はSC個52のバイトを置き換えるためのカウンタ変数である。S601では、変数SC値52を変数RL81に設定する。S602で変数RL値81が定数MARKLEN80より大きければ、S603で長さMARKLEN80に対する置換対象バイトを示すランレングス(RL)マーカを書き出し、S604で変数RL81から定数MARKLEN80を減じる。S602で変数RL値81が定数MARKLEN80より小さくなれば、最後にS605で残りの長さが変数RL値81の置換対象バイトを示すランレングス(RL)マーカを書き出し、S606で変数RL値81を0とする。

【0160】この処理を説明するフローチャートによれば、ランレングスマーカが分割されて連続出力されるとき最大長のものから出力され、最後に最大値未満のものが出力されることになるが、上記図27にも示したとおりこの出力順によらなくてもよい。また、置換対象バイトは必ずしも0x00または0xFFである必要はなく、符号出力を監視することによって任意の連続する符号バイトを置き換えてもよい。

【0161】ランレングスマーカのランレングス部のビット数が、SCカウンタ52のビット数以上であれば、ラン長の指示についてただ1つのランレングスマーカで指定することが可能である。

【0162】置換対象バイトは、必ずしもバイトであることはなく、例えばワード、ダブルワードといったマルチバイト単位でもよく、置き換えるランレングスマーカを適正に選択することによって、1バイトであってもよいし、複数から構成されるパターンを対象とすることもできる。

【0163】実施の形態7. 上記実施の形態6によれば、ラン長が変数SC値52の置換対象バイトはすべてランレングスマーカで置き換えられることになる。本実施の形態7では、図33において、一部のランレングス分だけランレングスマーカで置き換えずに置換対象バイトをその長さだけそのまま出力できるものとする。ここで、ランレングスマーカの構成バイト数を定数MARKLENMIN82とする。例えば、図28であれば定数MARKLENMINは2となる。

【0164】図34は、ランレングスマーカの置換え処理を行うRUNLENMARK処理を説明するフローチャートである。図34において、S601からS604およびS606は上記図32と同様の処理が行われる。S607において変数RL値81が定数MARKLENMIN82以上であれば、S605で長さが変数RL値81の置換対象バイトを示すランレングス(RL)マー

カを書き出し、定数MARKLENMIN82未満であればS608で置換対象バイトを変数RL値81で示される個数だけ書き出す。

【0165】このように、置換対象バイトのラン長がランレングスマーカの構成バイト数より少なければランレングスマーカへ置き換えずに置換対象バイトをそのランの長さ分だけそのまま出力することによって符号長を短くできる。どちらで出力しても長さが等しい場合には、置き換えても置き換えなくてもよい。

10 【0166】ただし、置換対象バイトがバイト0xFFである場合には、置き換えないうきSTUFFバイトの挿入が行われて符号長が2倍となるため、より短くなるように長さの判定を考慮すべきである。

【0167】実施の形態8. 上記実施の形態6あるいは上記実施の形態7において、ランレングスマーカを使用する際には、図35に示すように、出力される符号中に使用を通知する情報を含ませることができる。

20 【0168】常に使用する場合はランレングスマーカの使用通知は行う必要はないが、例えば上記図28から図30などのランレングスマーカの形式、置換対象バイトの種別、ランレングス部の長さの制限などの一部またはすべてについて、符号器から復号器へ付加情報を予め通知することによって動作モードの統一設定を図ることができる。

30 【0169】また、符号器は、符号化に先立って復号器に実装される処理能力を確認し(ネゴシエーション)、その結果から適用したモード情報を通知してもよい。復号器の能力としては、例えばランレングスマーカを逆変換するためのランレングス部のカウンタ桁数などの制限値がある。

【0170】さらに、この場合、復号器は、ランレングスマーカ使用通知を含まない符号データに対しては、従来の符号器の生成した符号としてみなすこともできる。また、符号器は、復号器が従来の復号器であれば、ランレングスマーカ使用通知を符号に挿入せず、またランレングスマーカを使用しなければよい。

40 【0171】実施の形態9. 上記実施の形態6あるいは上記実施の形態7に示されたランレングスマーカを従来の符号器および従来の復号器で使用するために、従来の符号器と回線(通信路)の間、または回線(通信路)と従来の復号器の間にアダプタを挿入することによって相互通信を可能とすることができる。

50 【0172】まず、図36に示すように、符号器側のアダプタ(ランレングスマーカ変換器)は、データから符号を区別し、符号バイトを監視しながら同一バイト値が連続する場合、そのラン長を改めてSCとすれば、図32または図34の処理を説明するフローチャートによってランレングスマーカへ置き換える。また、復号器側のアダプタ(ランレングスマーカ逆変換器)は、同様にデータから符号を区別し、符号バイトを監視しながらラン

レンジスマーカを検出した場合指示された置換対象バイトを検出されたラン長だけ連続して出力する。

【0173】また、図37に示すように、復号器側にランレンジスマーカ逆変換器がないとき、符号器はランレンジスマーカ変換器を実質的に使用せずにそのまま符号を通過させ、図38に示すように、符号器側にランレンジスマーカ逆変換器がないとき、復号器はランレンジスマーカ逆変換器を実質的に使用せずにそのまま符号を通過させることによってランレンジスマーカを使用できない従来装置との互換性を保持することができる。

【0174】ここで、従来の符号器／復号器に接続されるアダプタ（ランレンジ変換器／逆変換器）は、外見が符号器／復号器に対して独立構成であってもよいし、一体構成であっても構わない。

【0175】以上のように、符号化および復号処理は、計算機または装置の内部でデータを蓄積したり、公衆回線または専用回線による無線・有線による通信または記憶媒体によって、外部とのデータの交換を行うファクシミリ装置、スキャナ装置、プリンタ装置、コンピュータ、データベース装置、画像表示装置、画像蓄積装置、データ蓄積装置、画像伝送装置、データ伝送装置等のハードウェアに適用することができる。また、汎用の計算機にソフトウェアと一部のハードウェアで上記各装置相当の機能を実現することもでき、装置の外見が一体であっても複数の独立していても構わない。さらに、専用のハードウェア内部についても、これらの処理機能を搭載したLSI（半導体チップ）やミドルウェアによる実装形式には制限されるものではない。通信についても、電気的または光学的無線や有線、また公衆回線や専用回線、あるいはLAN、WAN、インターネット、イントラネットなどの通信形式に制限されるものではない。記憶媒体への記録形式も磁気的、電気的、光学的あるいはデジタル式やアナログ式といった記録形式に制限されるものでもなく、また、記録媒体は装置に固定されるか分離されてるかに制限されるものでもない。さらに、視認可否に関わらずインク等での記録形式でも構わない。

【0176】なお、この発明において、ブロック図、フロー図は画像符号化に適用するものとして説明を行ったが、一般的なデータの符号化にも用いることができることはいうまでもない。

【0177】

【発明の効果】以上のように、この発明によれば、符号が確定するまで待機する符号長を記憶するカウンタの桁あふれをなくすことができ、適当な間隔で符号を強制的に確定することにより桁上りを制御して符号の確定遅延時間を有限化して見積もることができる。また、符号値が確定したときに所定のパターンへの置き換えによって総符号長と符号の掃出し時間を短縮することができる。

【0178】また、桁上がり境界を厳密に計算せずに桁

上がり制御を行うことができると共に、復号器で符号器の符号レジスタを再現せずに桁上がり制御を行うことができるという効果が得られる。

【図面の簡単な説明】

【図1】 この発明の概念に係る上方／下方1／2領域切捨て方式の説明図である。

【図2】 この発明の実施の形態1に係る符号化部の構成を示すブロック図である。

【図3】 この発明の実施の形態1に係る復号化部の構成を示すブロック図である。

【図4】 この発明の実施の形態1に係るROUND DOWNE処理を説明するフローチャートである。

【図5】 この発明の実施の形態1に係るHALVING E処理を説明するフローチャートである。

【図6】 この発明の実施の形態1に係るROUND DOWND処理を説明するフローチャートである。

【図7】 この発明の実施の形態1に係るHALVING D処理を説明するフローチャートである。

【図8】 この発明の実施の形態1に係るHALVING E処理を説明するフローチャートである。

【図9】 この発明の実施の形態1に係るBYTE OUT 7処理を説明するフローチャートである。

【図10】 この発明の実施の形態1に係るHALVING D処理を説明するフローチャートである。

【図11】 この発明の実施の形態1に係るBYTE IN 7処理を説明するフローチャートである。

【図12】 この発明の実施の形態2に係る符号化部の構成を示すブロック図である。

【図13】 この発明の実施の形態2に係る復号化部の構成を示すブロック図である。

【図14】 この発明の実施の形態2に係るENCODER処理を説明するフローチャートである。

【図15】 この発明の実施の形態2に係るHALVING E処理を説明するフローチャートである。

【図16】 この発明の実施の形態2に係るCODE LOWER処理を説明するフローチャートである。

【図17】 この発明の実施の形態2に係るCODE UPPER処理を説明するフローチャートである。

【図18】 この発明の実施の形態2に係るDECODER処理を説明するフローチャートである。

【図19】 この発明の実施の形態2に係るHALVING D処理を説明するフローチャートである。

【図20】 この発明の実施の形態3に係るENCODER処理を説明するフローチャートである。

【図21】 この発明の実施の形態3に係るDECODER処理を説明するフローチャートである。

【図22】 この発明の実施の形態4に係るENCODER処理を説明するフローチャートである。

【図23】 この発明の実施の形態4に係るDECODER処理を説明するフローチャートである。

【図 24】 この発明の実施の形態 5 に係る ENCODER 処理を説明するフローチャートである。

【図 25】 この発明の実施の形態 5 に係る DECODER 処理を説明するフローチャートである。

【図 26】 この発明の実施の形態 6 に係るランレングスマーカの説明図である。

【図 27】 この発明の実施の形態 6 に係るランレングスマーカ分割出力の説明図である。

【図 28】 この発明の実施の形態 6 に係るランレングスマーカの一列を示す構成図である。

【図 29】 この発明の実施の形態 6 に係るランレングスマーカの一列を示す構成図である。

【図 30】 この発明の実施の形態 6 に係るランレングスマーカの一列を示す構成図である。

【図 31】 この発明の実施の形態 6 に係るバイト MK に置換対象パターンのバイト長を示し、パターンのラン長とともにパターンを直接指定する方法の説明図である。

【図 32】 この発明の実施の形態 6 に係る RUNLE NMARK 処理を説明するフローチャートである。

【図 33】 この発明の実施の形態 7 に係るランレングスマーカ符号長の比較に関する説明図である。

【図 34】 この発明の実施の形態 7 に係る RUNLE NMARK 処理を説明するフローチャートである。

【図 35】 この発明の実施の形態 8 に係るランレングスマーカ適用を通知する情報の説明図である。

【図 36】 この発明の実施の形態 9 に係るランレングスマーカを使用して符号器と復号器間の相互通信を行う構成の一例を示すブロック図である。

【図 37】 この発明の実施の形態 9 に係るランレングスマーカを使用して符号器と復号器間の相互通信を行う構成の一例を示すブロック図である。

【図 38】 この発明の実施の形態 9 に係るランレングスマーカを使用して符号器と復号器間の相互通信を行う構成の一例を示すブロック図である。

【図 39】 従来の算術符号化の概念を示す説明図である。

【図 40】 小数部を常に領域分割の精度に保ち、上位桁を整数部に掃き出すための従来の正規化処理の説明図である。

【図 41】 第 1 の従来例に係る符号器の構成を示すブロック図である。

【図 42】 第 1 の従来例に係る復号器の構成を示すブロック図である。

【図 43】 標準モデルテンプレートの説明図である。

【図 44】 確率推定テーブルの説明図である。

【図 45】 符号化／復号／領域幅レジスタの構成図である。

【図 46】 第 1 の従来例に係る ENCODER 処理を説明するフローチャートである。

【図 47】 第 1 の従来例に係る ENCODE 処理を説明するフローチャートである。

【図 48】 第 1 の従来例に係る CODELPS 処理を説明するフローチャートである。

【図 49】 第 1 の従来例に係る CODEMPS 処理を説明するフローチャートである。

【図 50】 第 1 の従来例に係る RENORME 処理を説明するフローチャートである。

【図 51】 第 1 の従来例に係る BYTEOUT 処理を説明するフローチャートである。

【図 52】 第 1 の従来例に係る INITENC 処理を説明するフローチャートである。

【図 53】 第 1 の従来例に係る FLUSH 処理を説明するフローチャートである。

【図 54】 第 1 の従来例に係る CLEARBITS 処理を説明するフローチャートである。

【図 55】 第 1 の従来例に係る FINALWRITES 処理を説明するフローチャートである。

【図 56】 第 1 の従来例に係る DECODER 処理を説明するフローチャートである。

【図 57】 第 1 の従来例に係る DECODE 処理を説明するフローチャートである。

【図 58】 第 1 の従来例に係る LPS_EXCHANGE 処理を説明するフローチャートである。

【図 59】 第 1 の従来例に係る MPS_EXCHANGE 処理を説明するフローチャートである。

【図 60】 第 1 の従来例に係る RENORMD 処理を説明するフローチャートである。

【図 61】 第 1 の従来例に係る BYTEIN 処理を説明するフローチャートである。

【図 62】 第 1 の従来例に係る INITDEC 処理を説明するフローチャートである。

【図 63】 第 2 の従来例に係る桁上がり／非桁上がり領域切捨て処理の説明図である。

【図 64】 第 2 の従来例に係る符号器の構成を示すブロック図である。

【図 65】 第 2 の従来例に係る復号器の構成を示すブロック図である。

【図 66】 第 2 の従来例に係る符号化／領域幅レジスタの構成図である。

【図 67】 第 2 の従来例に係る RENORME 処理を説明するフローチャートである。

【図 68】 第 2 の従来例に係る ROUNDDOWNE 処理を説明するフローチャートである。

【図 69】 第 2 の従来例に係る BYTEOUT 処理を説明するフローチャートである。

【図 70】 第 2 の従来例に係る INITENC 処理を説明するフローチャートである。

【図 71】 第 2 の従来例に係る FINALWRITES 処理を説明するフローチャートである。

【図 7 2】 第 2 の従来例に係る LPS_EXCHANGE 処理を説明するフローチャートである。

【図 7 3】 第 2 の従来例に係る RENORMD 処理を説明するフローチャートである。

【図 7 4】 第 2 の従来例に係る ROUNDDOWND 処理を説明するフローチャートである。

【図 7 5】 第 2 の従来例に係る BYTE IN 処理を説明するフローチャートである。

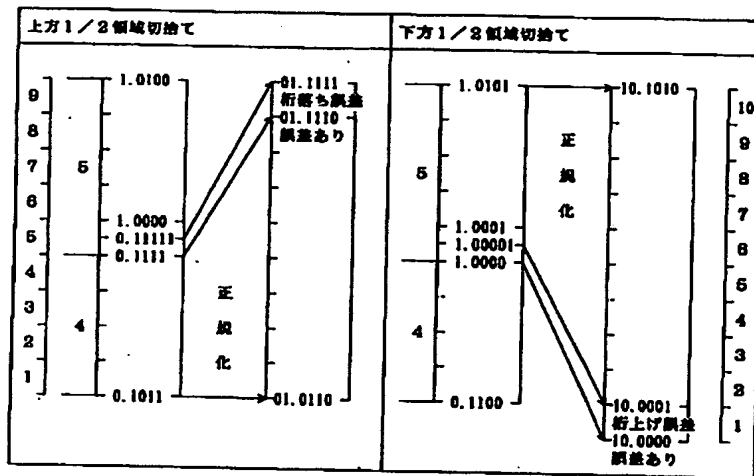
【図 7 6】 第 2 の従来例に係る INITDEC 処理を説明するフローチャートである。

【符号の説明】

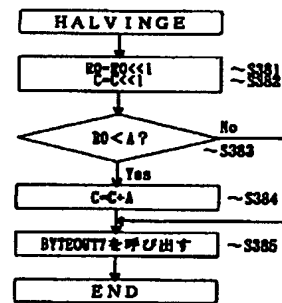
1 A 符号部、1 B 復号部、2 コンテキスト CX、3 画素 PIX、4 符号、5 データメモリ、6 画像、7 予測値テーブル MPS、8 ステートテーブル ST、9 LPS 領域幅テーブル LSZ、10 MPS 状態遷移先テーブル NMPS、11 LPS 状態遷移先テーブル NLPS、12 予測値反転判定テーブル SWITCH、13 A 算術符号器、13 B 算術復号器、1

4 シンボル、15 A 画素シンボル変換器、15 B シンボル画素変換器、16 予測値更新処理器、17 ステート更新処理器、18 ステートテーブル STATE、19 非予測値テーブル LPS、30 A 符号化レジスタ (下界値) C、30 B 復号レジスタ C、31 領域幅レジスタ A、32 Cx レジスタ、33 Cs レジスタ、34 Cb レジスタ、35 Cc レジスタ、36 CLOW レジスタ、37 Cb レジスタ、38 CHIGH レジスタ、39 Cx レジスタ、50 CT カウンタ、51 符号バッファ BUFFER、52 SC カウンタ、53 出力判定バッファ TEMP、60 上界値 U、61 桁上がり境界 T、62 桁上がり領域幅 R1、63 非桁上がり領域幅 R0、64 (復号器の) 符号レジスタ D、65 1/2 領域境界値 T、66 桁上がり境界値 CW、80 ランレングス部最大値 MARKLEN、81 ランレングスカウンタ RL、82 ランレングスマーカ長 MARKLENMIN。

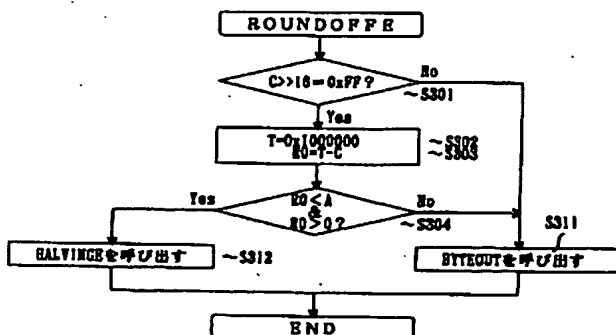
【図 1】



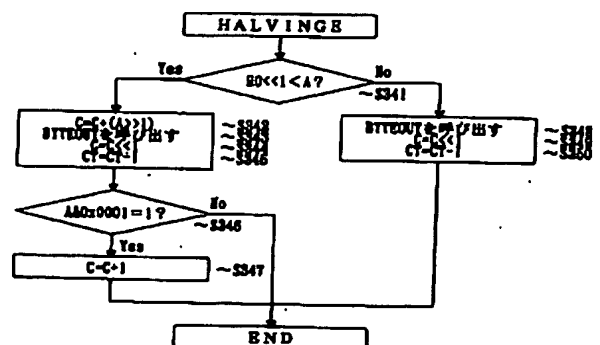
【図 8】



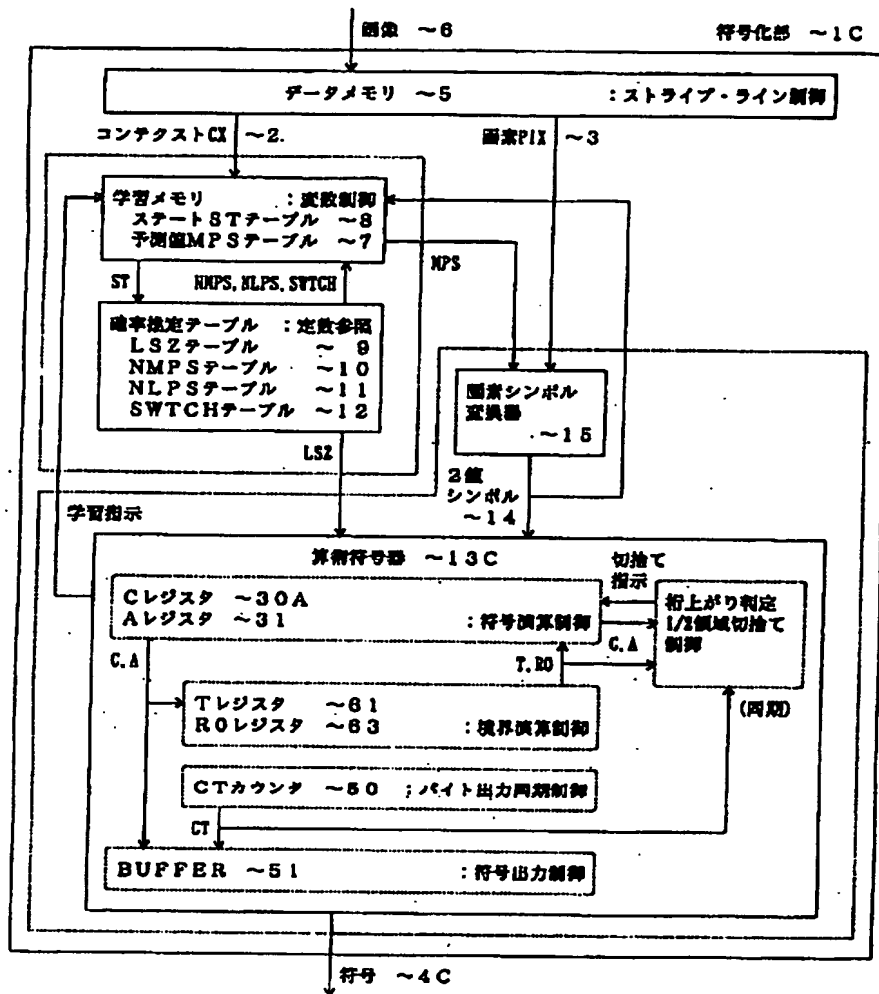
【図 4】



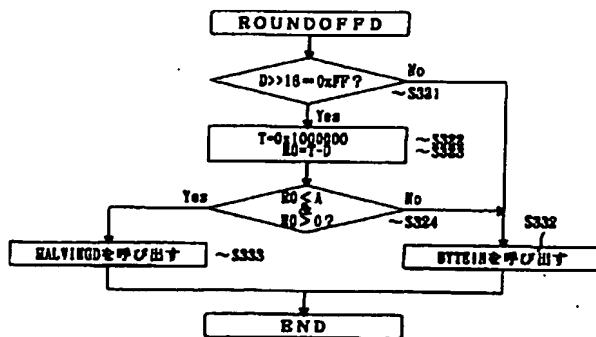
【図 5】



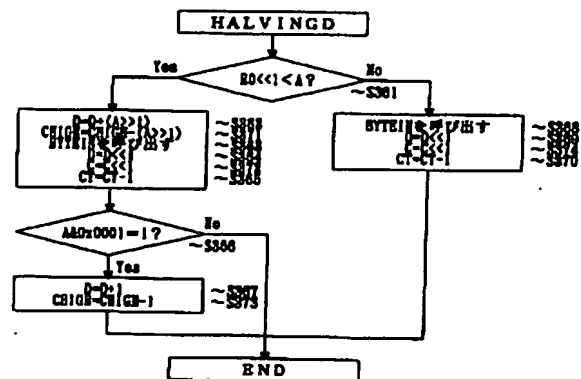
【図 2】



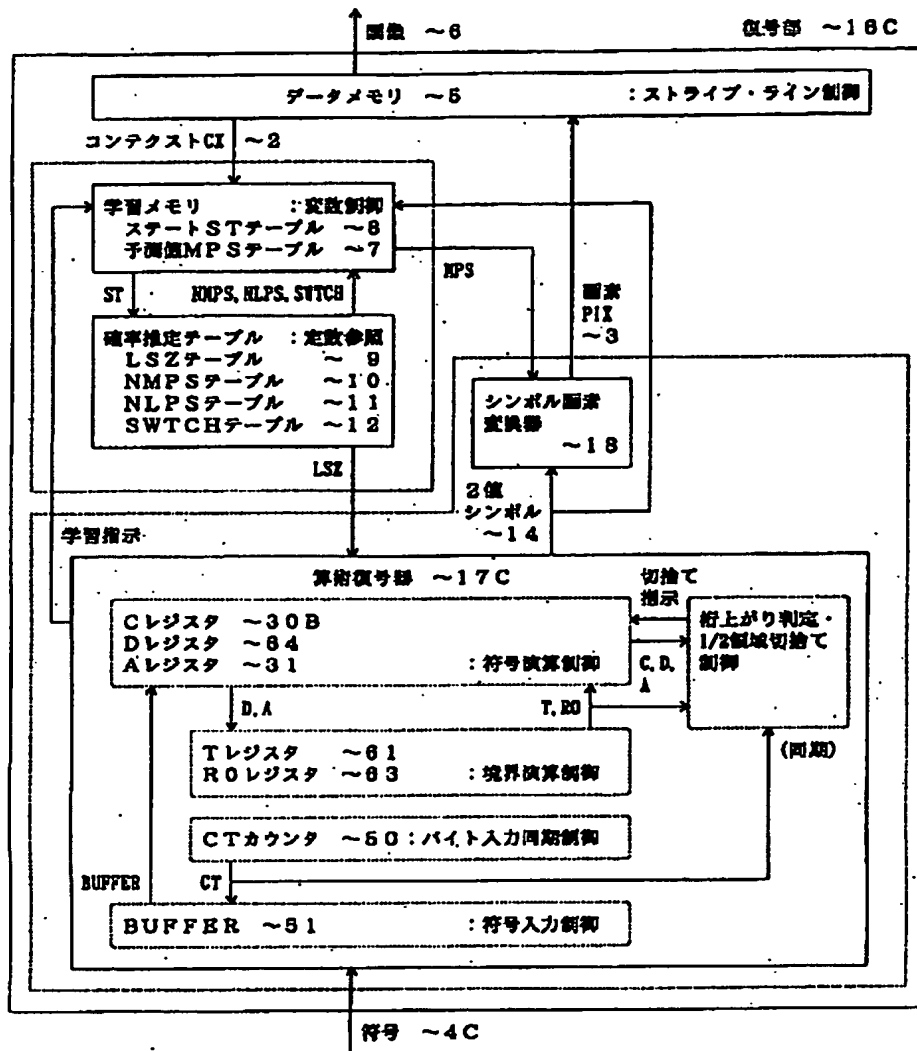
【図 6】



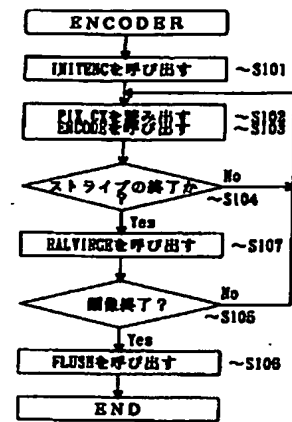
【図 7】



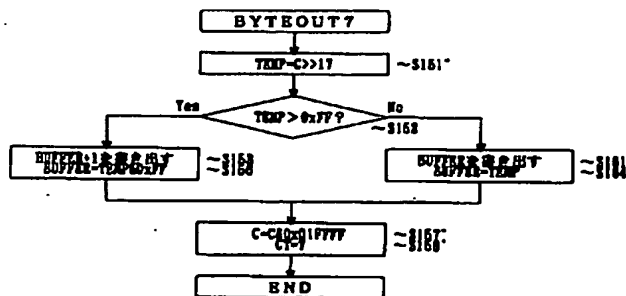
【図 3】



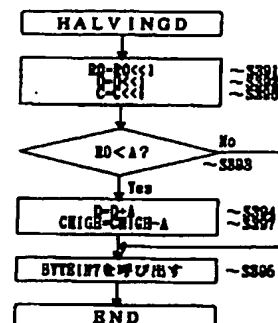
【図 14】



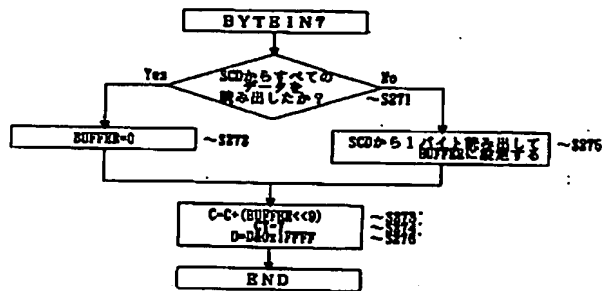
【図 9】



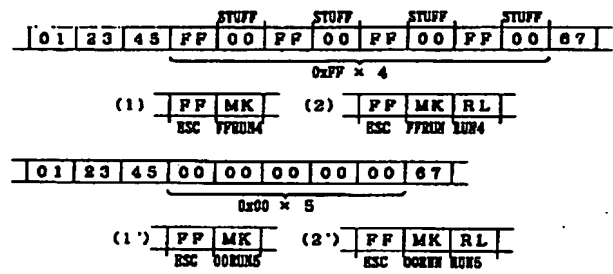
【図 10】



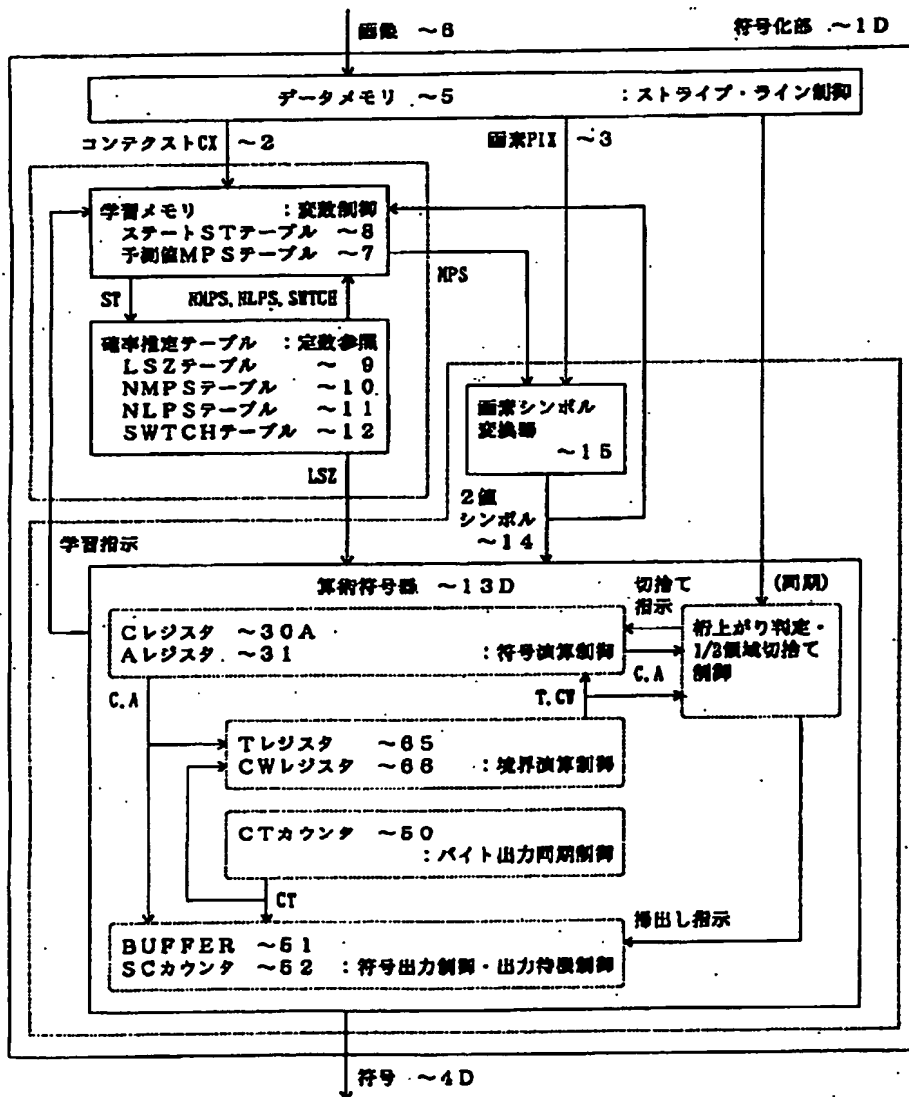
【図11】



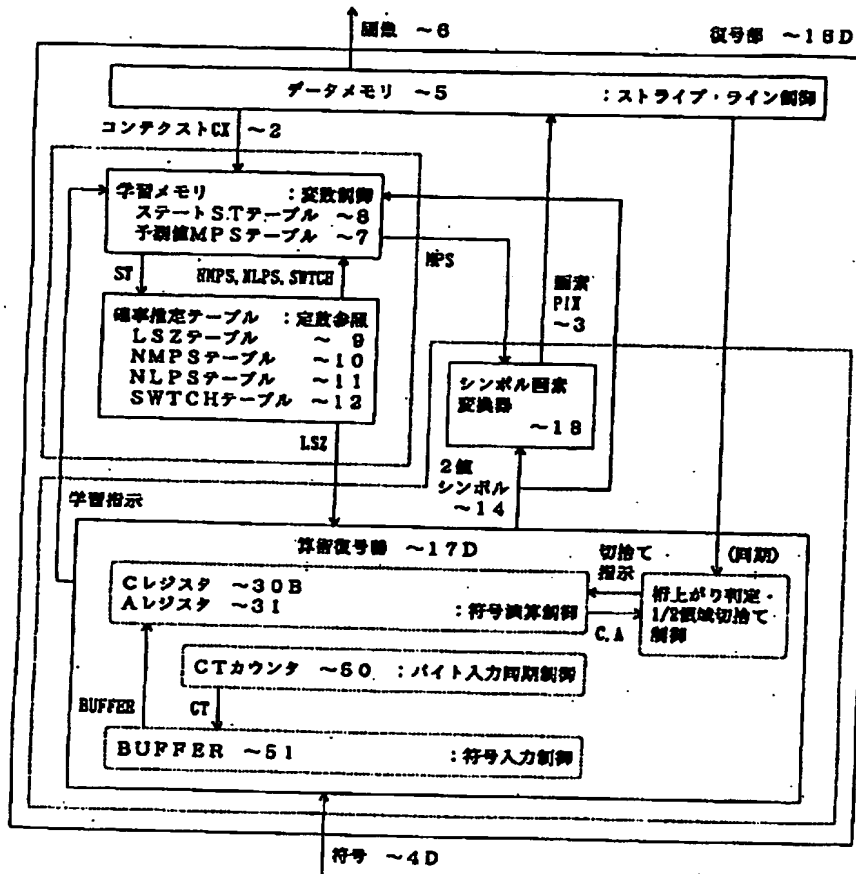
【図26】



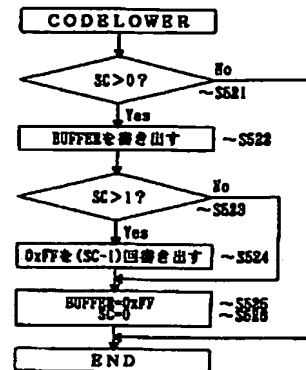
【図12】



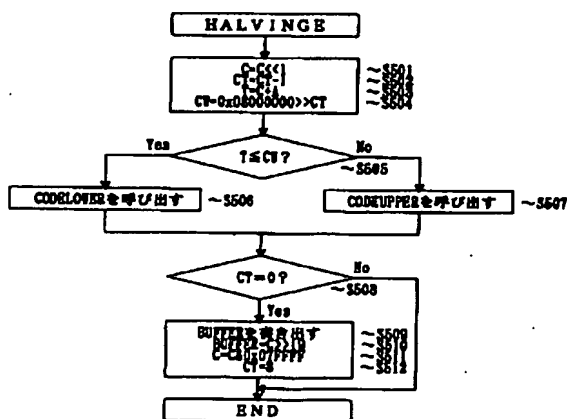
【図13】



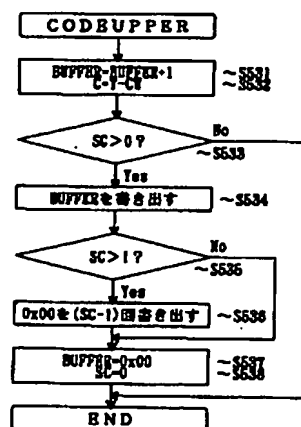
【図16】



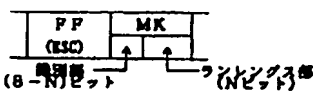
【図15】



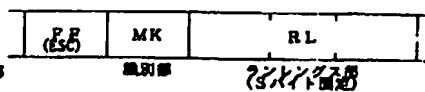
【図17】



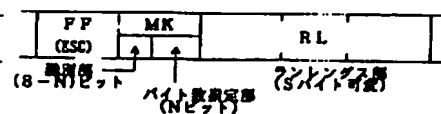
【図28】



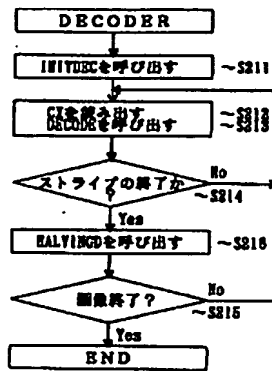
【図29】



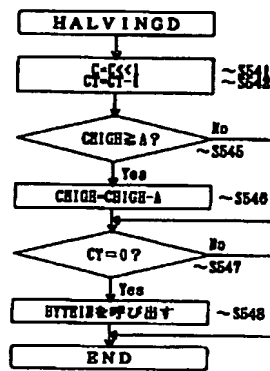
【図30】



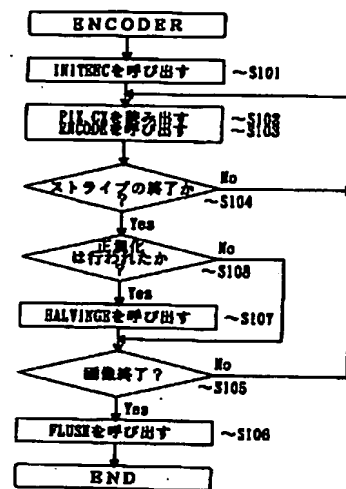
【図 18】



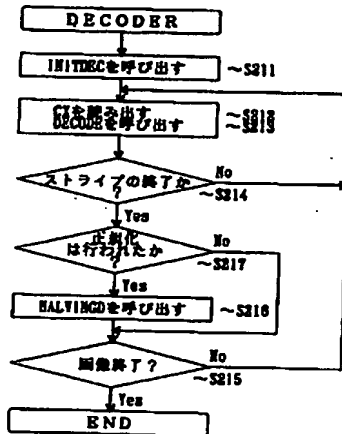
【図 19】



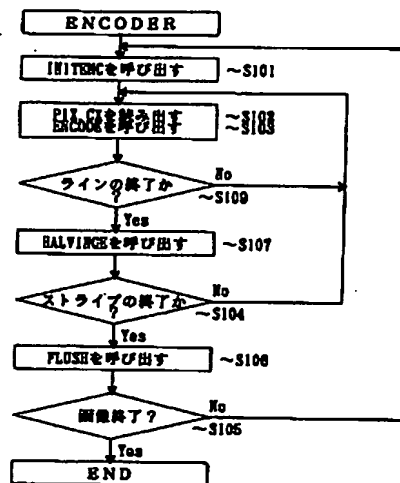
【図 20】



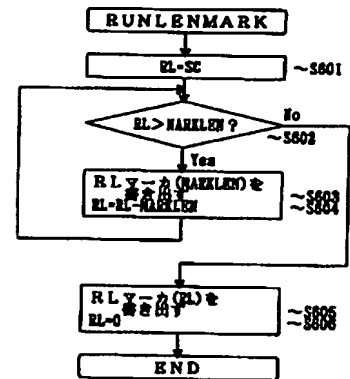
【図 21】



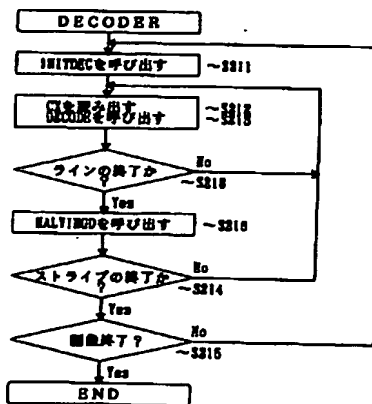
【図 22】



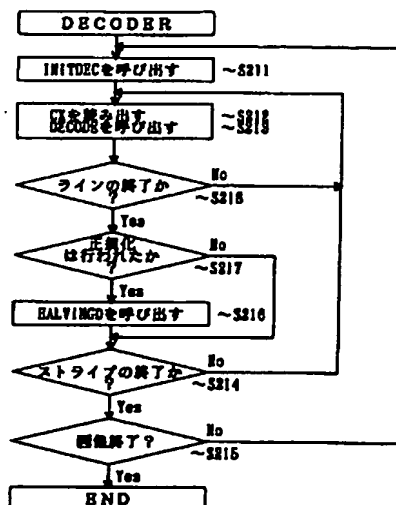
【図 32】



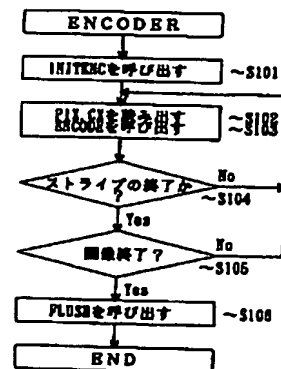
【図 23】



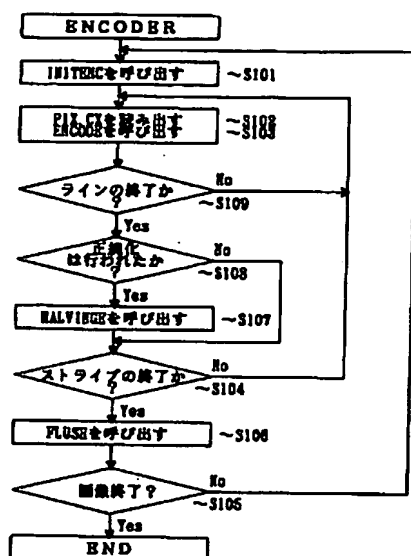
【図 25】



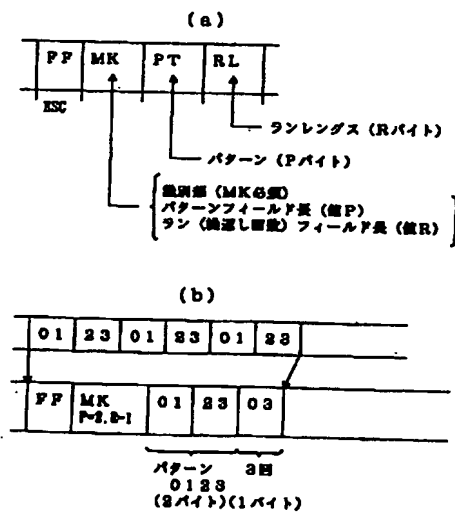
【図 46】



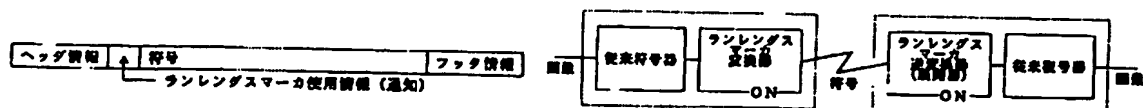
【图 24】



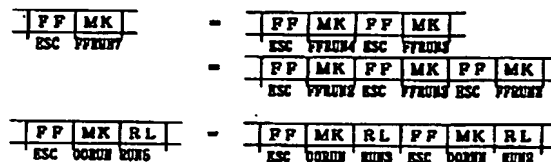
【图 3 1】



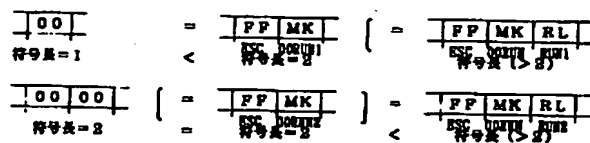
【图 3 5】



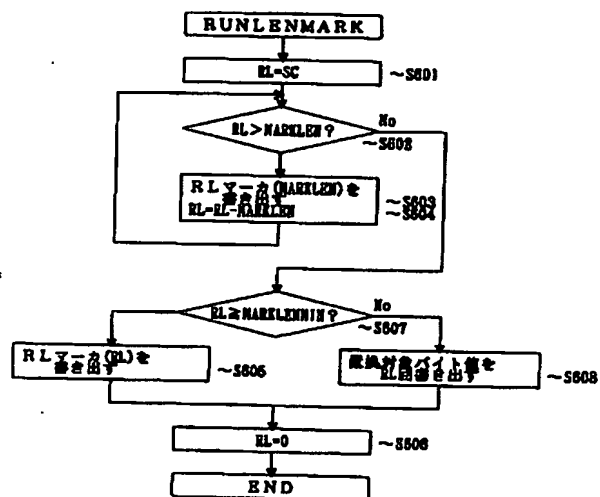
【图 2.7】



【图 3 3】



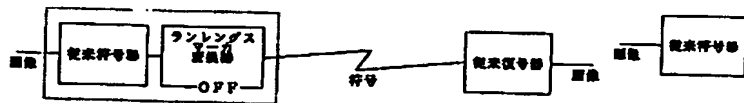
【图 3 4】



【图 3 6】



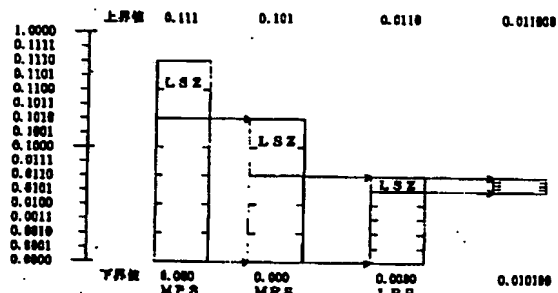
【图 3 7】



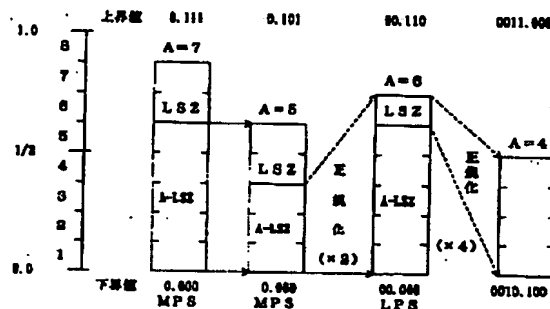
【图 3 8】



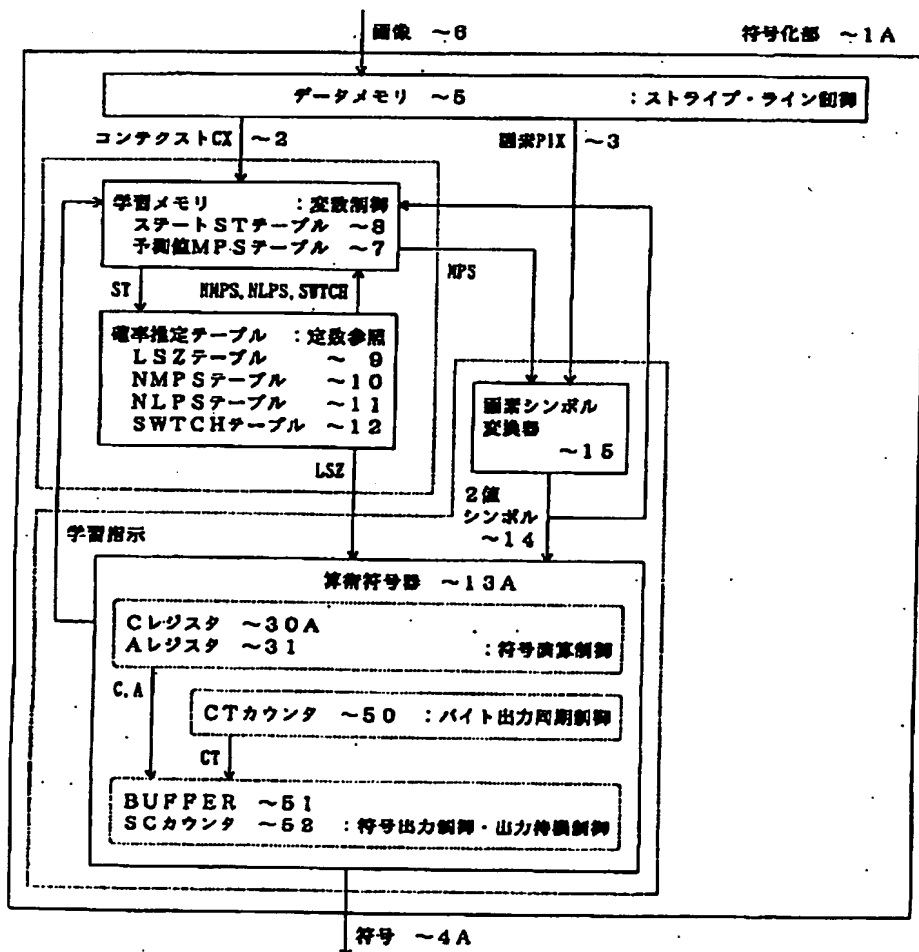
【図 3 9】



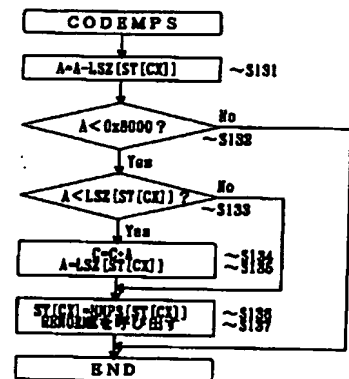
【図 40】



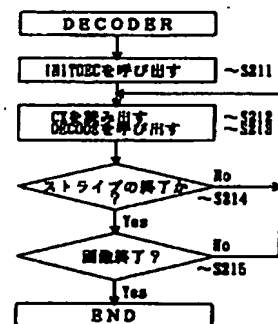
【图 4 1】



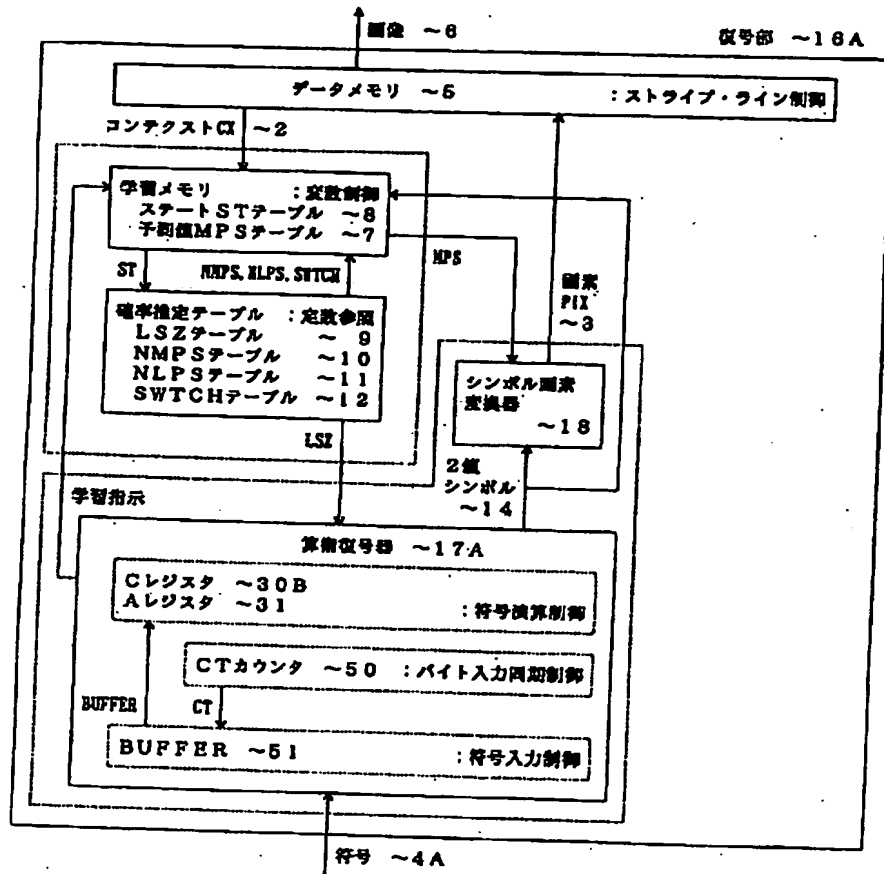
【图 4 9】



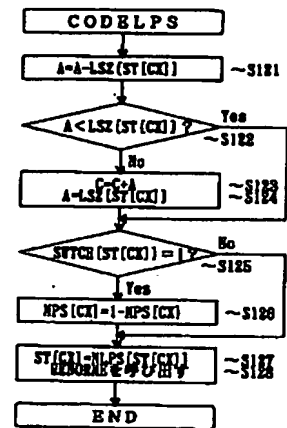
【图 5 6】



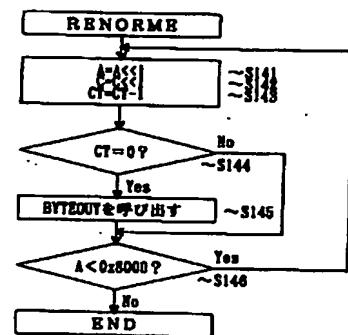
【図42】



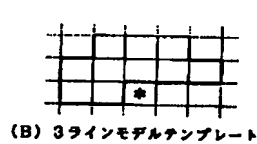
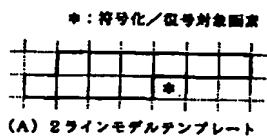
【図48】



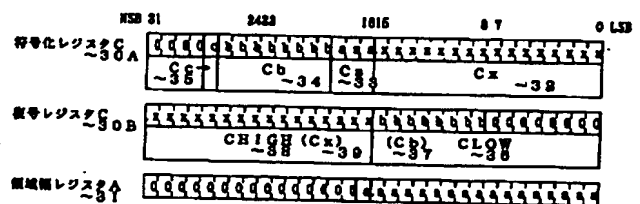
【図50】



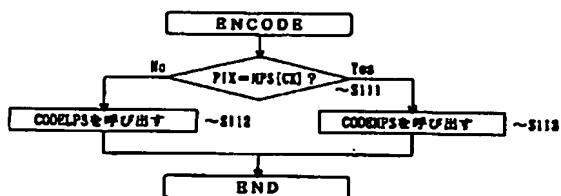
【図43】



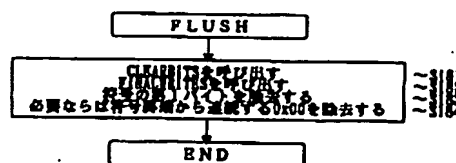
【図45】



【図47】



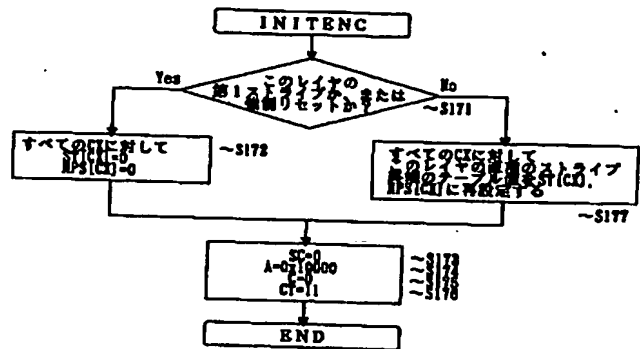
【図53】



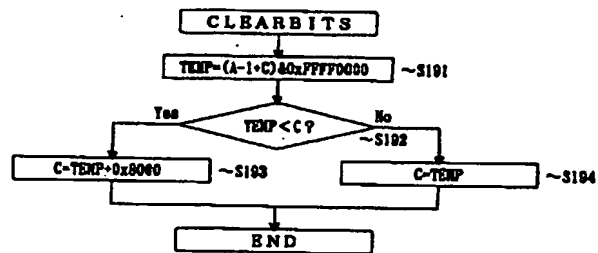
【図 44】

ST	LSX	MPSP	MPSP	SWTCH	ST	LSX	MPSP	MPSP	SWTCH
0	0x512d	1	1	0	47	0x01ad	53	98	0
1	0x2585	16	2	0	50	0x0160	56	59	0
2	0x1112	16	4	0	50	0x0173	57	60	0
3	0x080d	16	4	0	50	0x0078	58	61	0
4	0x03d8	20	5	0	41	0x00eb	59	62	0
5	0x01da	23	6	0	63	0x008f	61	32	0
6	0x00a5	25	7	0	63	0x008f	61	32	0
7	0x006f	28	8	0	64	0x5b12	65	65	1
8	0x0036	30	9	0	65	0x4d04	80	66	0
9	0x001a	31	10	0	66	0x417c	81	67	0
10	0x000d	35	11	0	67	0x37d8	82	68	0
11	0x0006	9	12	0	68	0x2f08	83	69	0
12	0x0003	10	13	0	69	0x293c	82	70	0
13	0x0001	12	13	0	70	0x2379	86	71	0
14	0x5a72	15	15	0	71	0x1a05	87	72	0
15	0x3f25	16	16	0	72	0x1a05	87	72	0
16	0x2cf2	18	17	0	73	0x174a	72	74	0
17	0x207c	19	18	0	74	0x1424	72	75	0
18	0x17b9	20	19	0	75	0x119c	74	76	0
19	0x11f2	22	20	0	76	0x0f65	74	77	0
20	0x0ca7	21	21	0	77	0x0451	75	78	0
21	0x09a1	25	22	0	78	0x0bb5	77	79	0
22	0x032f	28	23	0	79	0x0400	77	80	0
23	0x055c	28	24	0	80	0x5832	80	81	1
24	0x0405	29	25	0	81	0x4d1c	88	82	0
25	0x0303	31	26	0	82	0x438a	89	83	0
26	0x0240	32	27	0	83	0x3bd4	90	84	0
27	0x01b1	34	28	0	84	0x34aa	91	85	0
28	0x0146	36	29	0	85	0x2aa9	92	86	0
29	0x00f4	37	30	0	86	0x259a	93	87	0
30	0x00b7	39	31	0	87	0x5115	88	71	0
31	0x008a	40	32	0	88	0x5570	88	71	0
32	0x0068	42	33	0	89	0x4ca2	95	80	0
33	0x004e	43	34	0	90	0x44d8	96	81	0
34	0x003b	44	35	0	91	0x3a22	97	82	0
35	0x002c	45	36	0	92	0x3824	99	83	0
36	0x001a	47	37	1	93	0x32b4	99	84	0
37	0x0000	48	38	1	94	0x2a17	93	85	0
38	0x3a04	49	39	0	95	0x56a8	95	86	1
39	0x2ef1	49	40	0	96	0x4466	101	92	0
40	0x261f	50	41	0	97	0x4753	102	93	0
41	0x1f33	52	42	0	98	0x41ef	103	94	0
42	0x19a8	53	43	0	99	0x3bd4	104	100	0
43	0x1518	54	44	0	100	0x375a	99	93	0
44	0x1177	55	45	0	101	0x5231	105	102	0
45	0x0a74	56	46	0	102	0x4a07	106	103	0
46	0x0b7b	57	47	0	103	0x4639	107	104	0
47	0x09f8	58	48	0	104	0x4158	107	104	0
48	0x0861	59	49	0	105	0x5637	105	105	1
49	0x0706	59	50	0	106	0x5077	108	102	0
50	0x05cd	60	51	0	107	0x4b85	109	103	0
51	0x04de	60	52	0	108	0x5597	110	109	0
52	0x040f	60	53	0	109	0x5047	111	107	0
53	0x0363	61	54	0	110	0x5100	110	111	1
54	0x02d4	62	55	0	111	0x5327	112	108	0
55	0x025c	63	56	0	112	0x55a0	112	111	1
56	0x01f8	64	57	0					

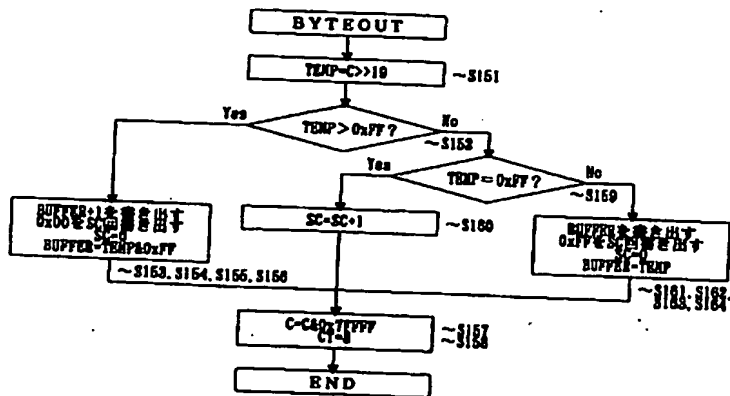
【図 52】



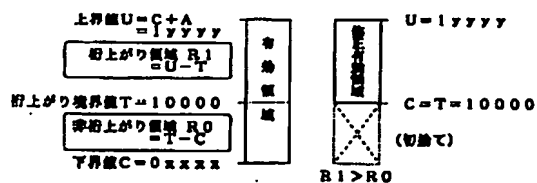
【図 54】



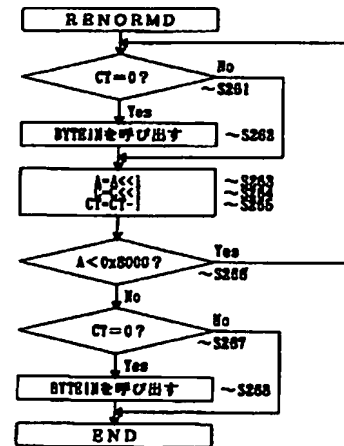
【図 51】



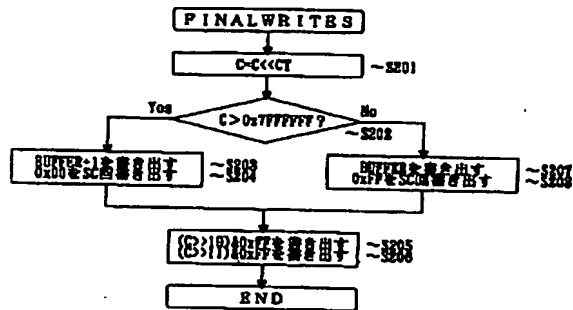
【図 63】



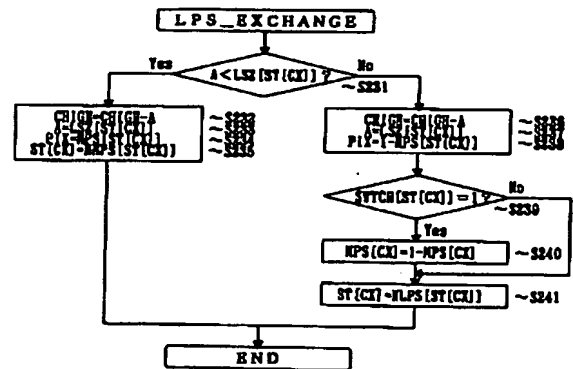
【図 60】



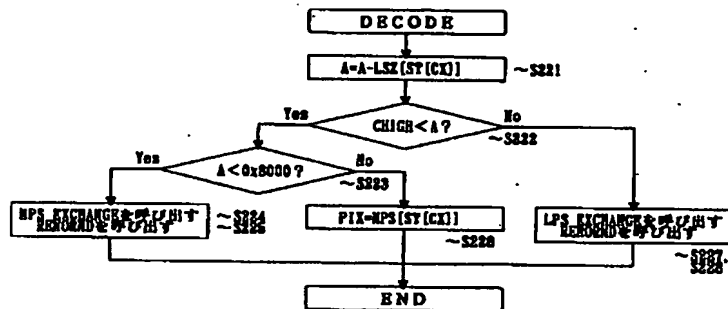
【図 55】



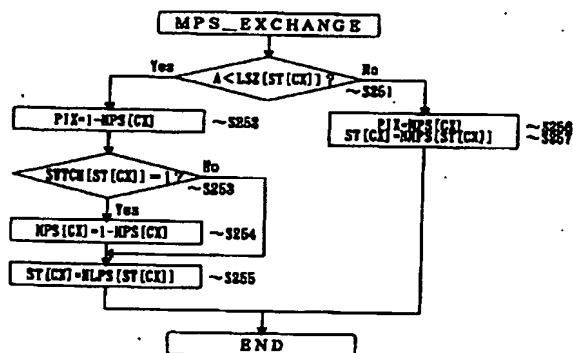
【図 58】



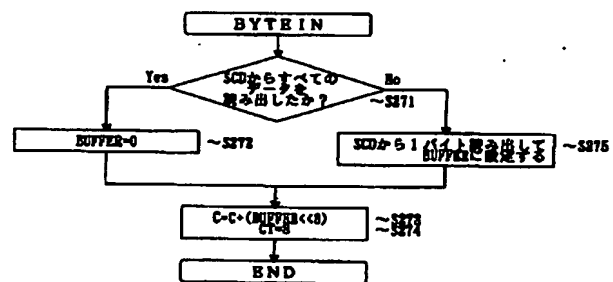
【図 57】



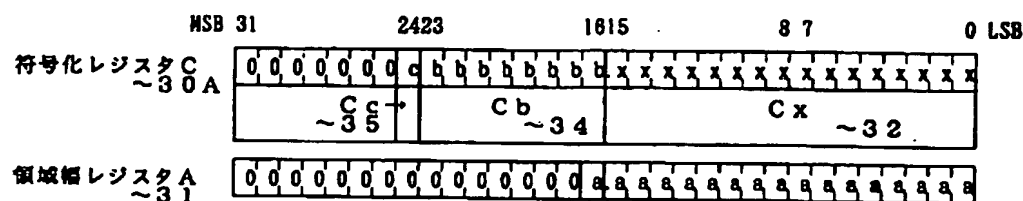
【図 59】



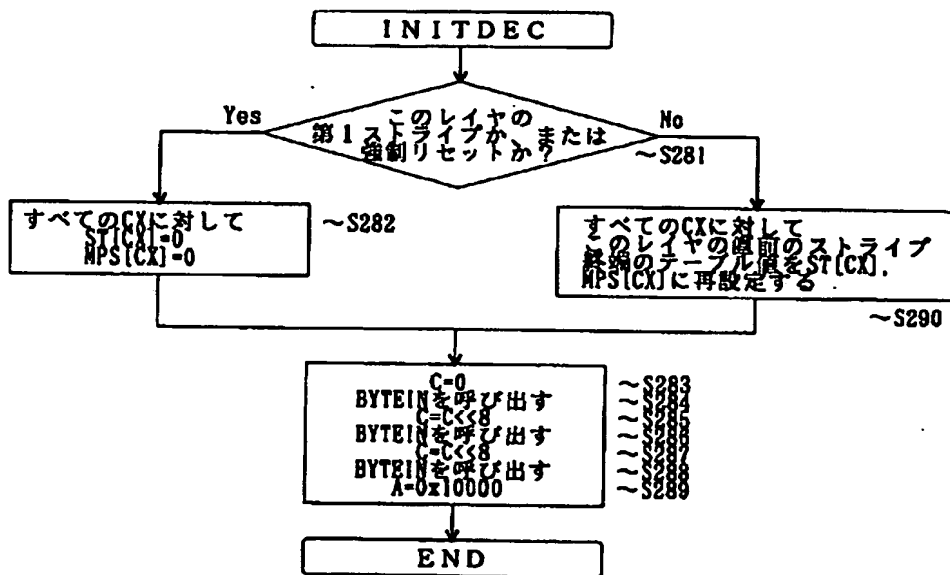
【図 61】



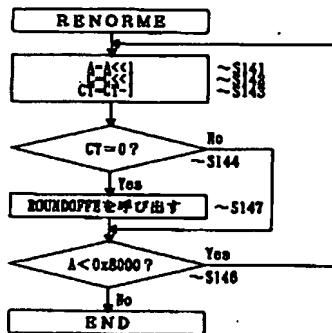
【図 66】



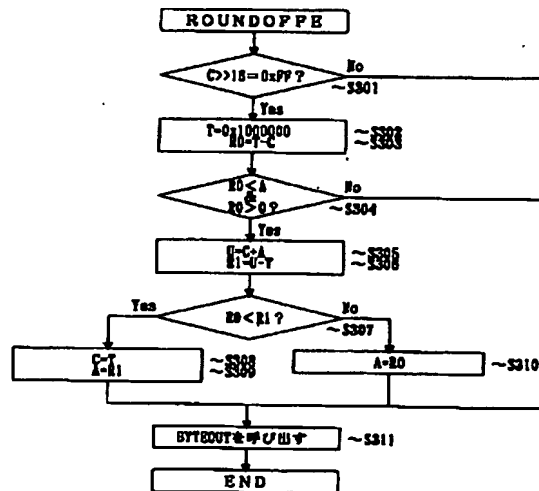
【図 62】



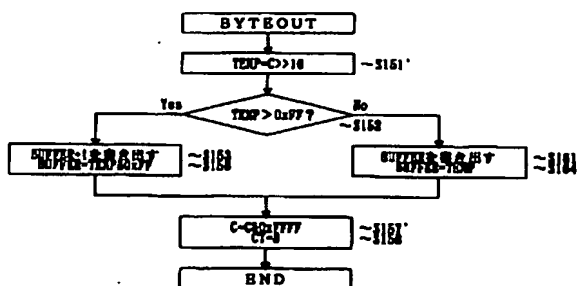
【図 67】



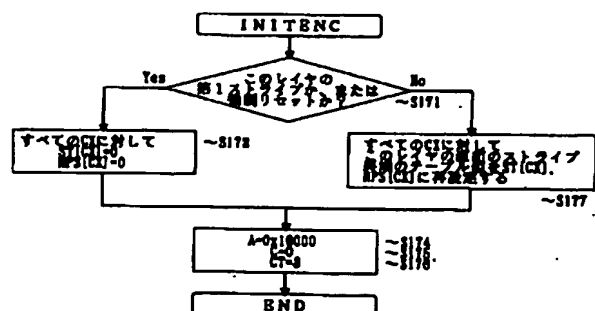
【図 68】



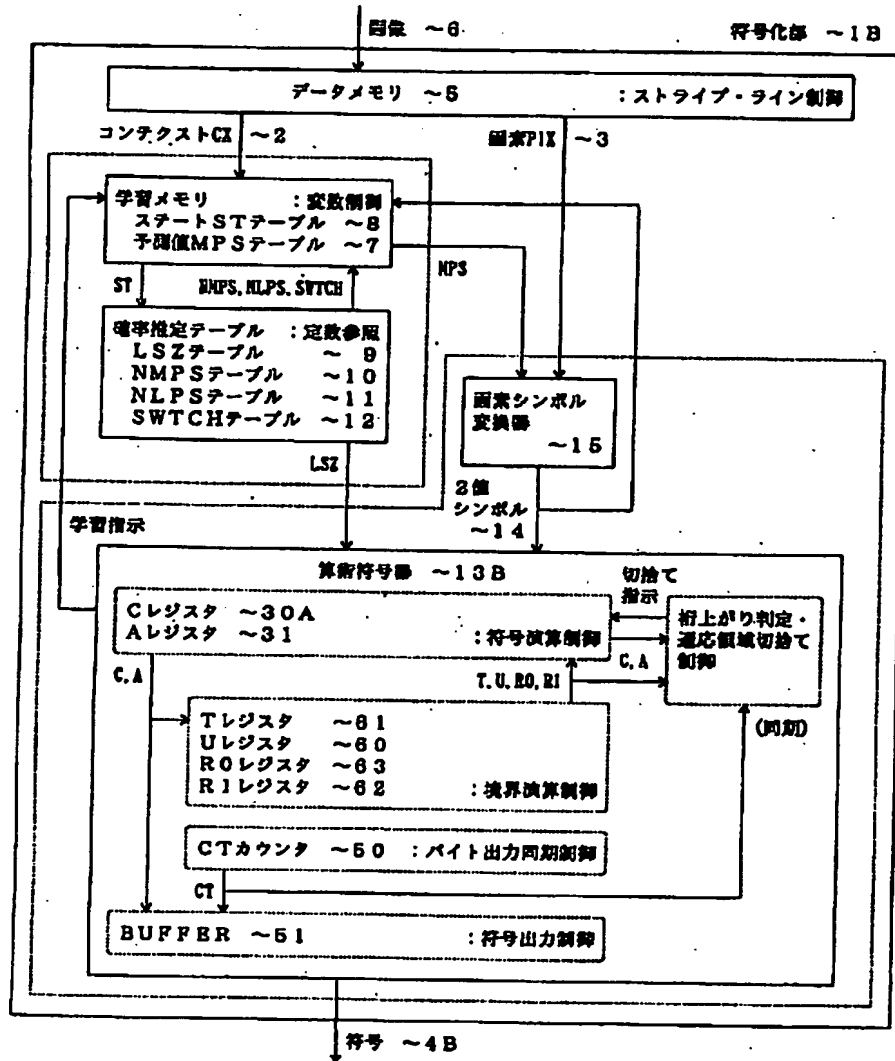
【図 69】



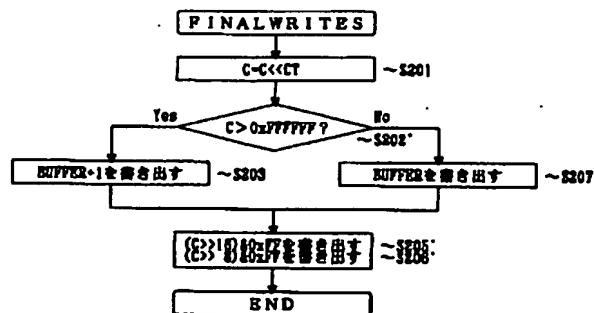
【図 70】



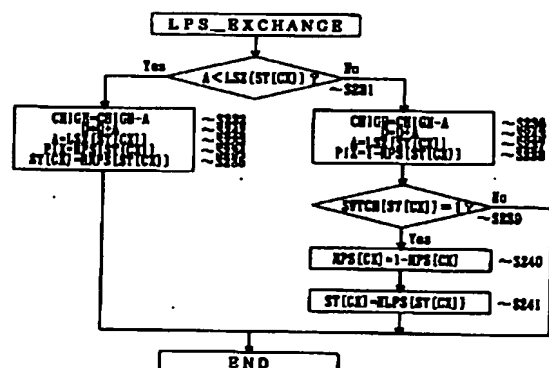
【図 64】



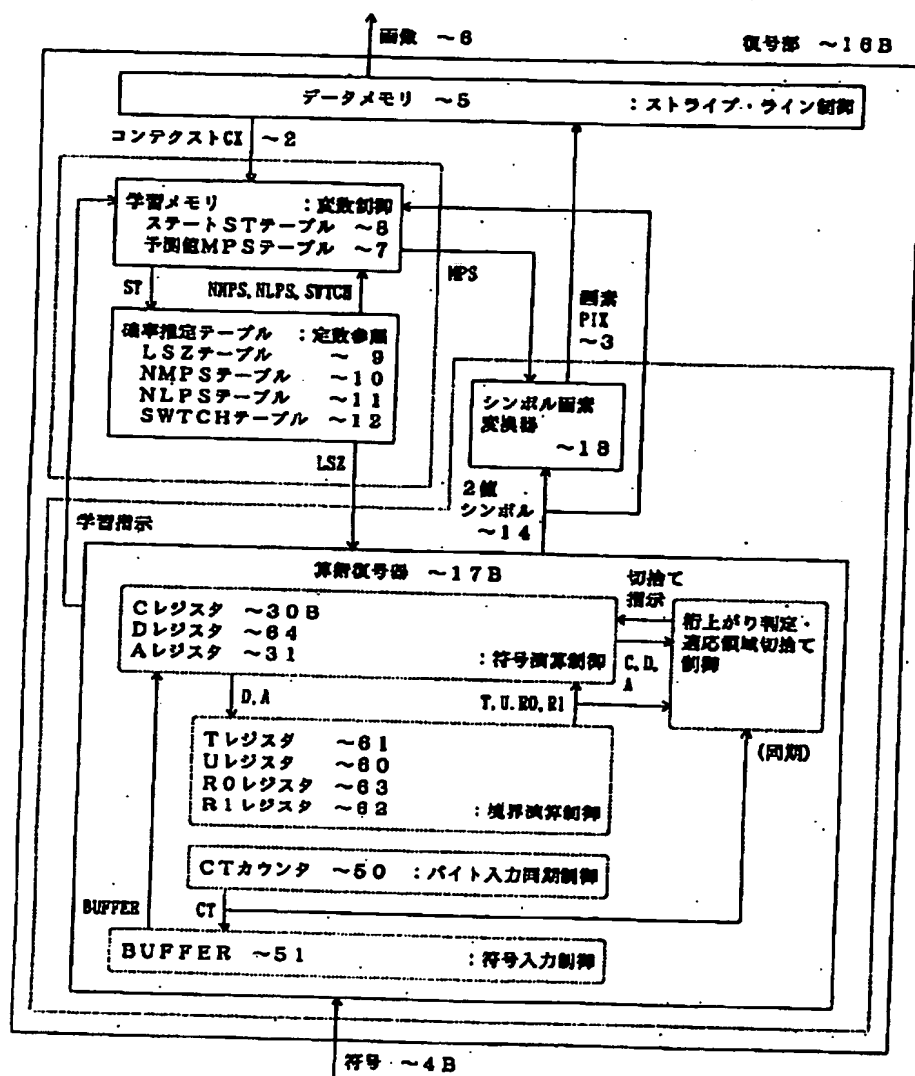
【図 71】



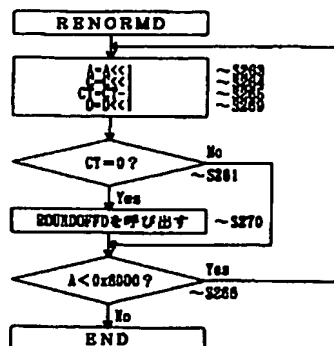
【図 72】



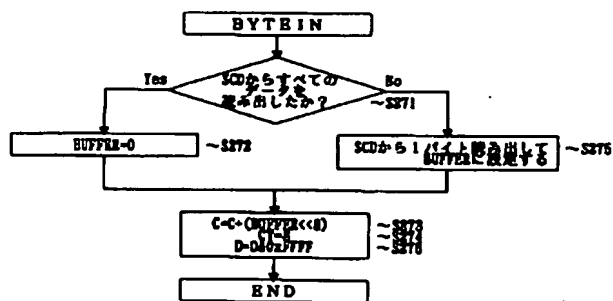
【图 6 5】



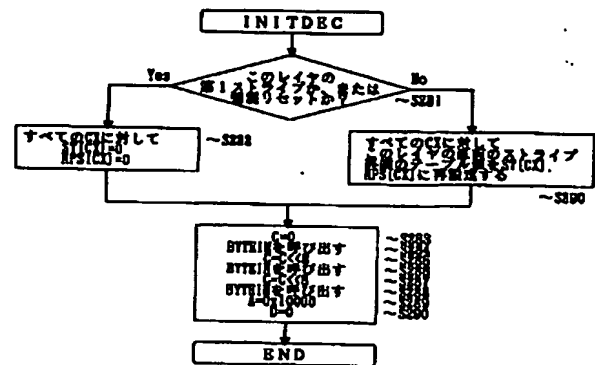
【图 7 3】



【图 7 5】



【图 7 6】



Fターム(参考) 5C059 KK15 ME05 ME11 UA02 UA05
UA34 UA38
5C078 BA21 CA31 DA00 DA01 DA02
DA06 DA11
5J064 AA02 BA08 BA10 BB09 BB12
BC01 BC02 BC04 BC14 BC17
BC28
9A001 BB03 EE04 HH05 HH27 JJ71
KK54